Real Time Kinematic Positioning Aiding with 3D Surface Model and Fisheye Camera

Ville Joensuu

School of Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 2.2.2017

Thesis supervisor:

Prof. Martin Vermeer

Thesis advisor:

M.Sc. Esa-Pekka Sundell

Aalto University
School of Engineering

AALTO UNIVERSITY
SCHOOL OF ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Ville Joensuu

Title: Real Time Kinematic Positioning Aiding with 3D Surface Model and Fisheye Camera

Date: 2.2.2017 Language: English Number of pages: 7+89

Department of Built Environment

Professorship: Geodesy

Supervisor: Prof. Martin Vermeer

Advisor: M.Sc. Esa-Pekka Sundell

This study presents an approach to obstruction masking for GNSS Real Time Kinematic (RTK) positioning using information from 3D models and sky-view images. The signals propagating from obstructed satellites to the receiver are one of the largest error sources in RTK positioning.

The effects of signal masking were assessed by conducting a test campaign in varying environments. The data recorded in the test campaign was used to compare the results of aided RTK solutions to unaided ones. From the results of the tests it was found that combining image and 3D model masking is the optimal mask generation method providing the most significant improvement in the RTK positioning performance.

The effects of signal masking were found to depend on the used processing software, with the largest performance improvement in terms of positioning accuracy and fix availability obtained when using open-source RTKLIB processing software and in lesser extent when using commercial Septentrio PPSDK RTK processing software. The commercial feasibility of such system consequently depends on the used RTK processing algorithms.

Keywords: GNSS, RTK, NLOS reception, multipath, satellite masking, obstruction masking

AALTO-YLIOPISTO
INSINÖÖRITIETEIDEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Ville Joensuu

Työn nimi: Työn nimen suomennos: Real Time Kinematic paikannuksen avustaminen 3D pintamallilla sekä taivaskameralla

Päivämäärä: 2.2.2017 Kieli: Englanti Sivumäärä: 7+89

Maankäyttötieteiden laitos

Professuuri: Geodesia

Työn valvoja: Prof. Martin Vermeer

Työn ohjaaja: DI Esa-Pekka Sundell

Tässä tutkimuksessa esitellään menetelmä esteiden peittämien satelliittien suodattamiseen GNSS Real Time Kinematic (RTK) -paikannuksesta käyttämällä hyödyksi 3D-malleja sekä taivaasta otettavia valokuvia. Esteiden peittämien satelliittien signaalit saattavat kulkeutua satelliiteista vastaanottimeen heijastumalla vastaanotinta ympäröivistä pinnoista.
Satelliittien suodattamisen vaikutuksia paikannuksen laadulle selvitettiin toteuttamalla testimittauskampanja vaihtelevissa satelliittipaikannusympäristöissä. Testeissä kerättyä dataa analysoitiin vertailemalla ilman sateliittisuodatusta, sekä satelliittisuodatuksen kanssa prosessoituja paikannusratkaisuja. Analyysin tulosten perusteella havaittiin, että optimaalisin keino hyödyntää satelliittipaikannusympäristöstä testijärjestelmällä kerättyä tietoa oli yhdistää sekä 3D-mallilta, että kuvista saatu tieto. Kuvien sekä 3D-mallien keräämän tiedon yhdistäminen satelliittisuodatukseen tuotti merkittävimmän parannuksen RTK-paikannuksen laatuun.
Suodatusmenetelmän vaikutusten havaittiin yhtäältä riippuvan myös RTK-prosessointiin käytetystä ohjelmistosta. Avoimen lähdekoodin RTKLIB ohjelmistolla prosessoitaessa RTK-paikannuksen tarkkuus ja saatavuus paranivat merkittävästi, kun taas kaupallisella Septentrio PPSDK -ohjelmistolla prosessoitaessa vaikutukset paikannuksen suorituskykyyn olivat vähäisempiä. Kaupallisen satelliittisuodatusjärjestelmän kannattavuus riippuukin RTK prosessointiin käytetyistä algoritmeista.

Avainsanat: GNSS, RTK, monitieheijastus, satelliittien suodatus

Preface

This thesis is written for Finnish systems and software engineering company SSF as a part of the Intelligent Signal Masking for GNSS RTK (ISMASK) research project. SSF, also known as Space Systems Finland, has a vast experience working with GNSS technologies from development of pseudolite systems to development of Galileo GNSS. The thesis is written under supervision of Esa-Pekka Sundell, a SSF employee with history especially in development of pseudolite systems. I want to thank SSF for the opportunity to participate in the ISMASK project and write this thesis.

I want to thank especially my instructor Esa-Pekka Sundell and my supervisor Professor Martin Vermeer for their good guidance.

I want also to thank other people who have participated in the project and of course my awesome wife.

Finally, I want to thank my fellow students known as *Geodesian Urhoolliset Uurtajat* Anton, Henri, Topi, and Tuukka for their marvelous help, support, and most of all sharing the pain of being a student.

Otaniemi, 2.2.2017

Ville Joensuu

Contents

Symbols and abbreviations

Notation

Matrices are denoted by uppercase boldface, e.g. matrix $\mathbf{M}$. Vectors are denoted by lowercase boldface, e.g. vector $\boldsymbol{\rho}$.

Symbols

c	speed of light in vacuum $\approx 3 \times 10^8$ m/s
δ	delta
Δ	Delta

Operators

$\sum_{i=0}^{n}$	sum over index i from 0 to n
$\left\|\mathbf{x}\right\|$	Euclidean norm of $\mathbf{x}$
$\int_a^b f(x)dx$	Integral of function f with respect to x over interval $[a, b]$

Abbreviations

APME	A Posteriori Multipath Estimation
BPSK	Binary Phase-shift Keying
CORS	Continuously Operating Reference Station
COTS	Commercial-of-the-Shelf
DD	Double Difference
DGNSS	Differential Global Navigation Satellite System
DLL	Delay Lock Loop
DMI	Distance Measurement Instrument
DOP	Dilution of Precision
DSLR	Digital Single-lens Reflex
ESA	European Space Agency
ESTEC	European Space Research and Technology Centre
FDE	Fault Detection and Exclusion
FLL	Frequency Lock Loop
GAMS	GPS Azimuth Measurement System
GDOP	Geometry Dilution of Precision
GEO	Geostationary Orbit
GIS	Geographic Information System
GLONASS	Global Navigation System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSO	Geosynchronous orbit
HDOP	Horizontal Dilution of Precision
I	In-phase
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ISMASK	Intelligent Signal Masking for GNSS RTK
LOD	Level of Detail
MEDLL	Multipath Estimating Delay Lock Loop
MEO	Medium Earth Orbit
NTP	Network Time Protocol
OTF-AR	On the Fly Ambiguity Resolution
PPSDK	Post Processing Software Development Kit
PDOP	Position Dilution of Precision
PLL	Phase Lock Loop
PPP	Precise Point Positioning
PVT	Position, Velocity and Time
Q	Quadrature
RAIM	Receiver Autonomous Integrity Monitoring
RF	Radio Frequency
RMS	Root Mean Square
RTK	Real Time Kinematic
SLIC	Simple Linear Iterative Clustering
SPS	Standard Positioning Service
TDOP	Time Dilution of Precision
USB	Universal Serial Bus
VDOP	Vertical Dilution of Precision
WGS-84	World Geodetic System 1984

1 Introduction

Global Navigation Satellite System (GNSS) technologies are nowadays widely used in almost all sectors of society. Satellite based positioning has enabled easily obtainable positioning for the masses with accuracy ranging from meters to tens of meters. For the more demanding usage purposes GNSS offers more accurate and robust positioning with classical *differential GNSS* (DGNSS), *real time kinematic* (RTK), *precise point positioning* (PPP), and other similar techniques. These enhanced processing techniques are used for many professional applications in fields from aviation to surveying.

Positioning—the main usage purpose of GNSS—is the art of determining the location of an object relative to other known locations. Positioning can be done relative to a known point or multiple known points. The known reference point can be for example the mass center of the Earth. The reference points are an essential requirement for positioning. [6, pp. 3–4] In this thesis, navigation refers to a concept of determining position and orientation of a navigating body taking the concept of positioning a step further.

The goal of positioning and navigation is to obtain a solution—in the case of GNSS positioning a positioning solution and in the case of navigation a navigation solution. The solution is the output of the used positioning or navigation system, a GNSS receiver for example. A positioning solution consists of the user's coordinates and possibly velocity and time estimates as well which is the case of the *position, velocity, time* (PVT) solution often output by GNSS receivers. A navigation solution includes velocity and attitude in addition to the user coordinates. A GNSS solution includes often a time component as the GNSS processing requires solving the accurate time in addition the position. Consequently, one of the most important usage purposes of GNSS besides positioning is to provide accurate timing information for many sectors of society.

In order to provide a positioning solution, the GNSS receiver has to process the data received from satellites. In optimal circumstances the receiver could utilize all satellites it is able to receive but sometimes one or more of these satellite signals might be erroneous leading to degradation of positioning performance. In order to mitigate the errors caused by erroneous satellites, the receiver may try to select only satellites it considers of sufficient quality to the processing. These kinds of satellite selection methodologies are often implemented in GNSS receivers and processing software. One commonly used satellite selection algorithm type is *receiver autonomous integrity monitoring* (RAIM) [11].

As a satellite based navigation system GNSS utilizes signals transmitted from the satellites to the user receiver. In order to provide a good quality solution, the receiver antenna requires an unobstructed visibility of the satellites. In obstructed environments, such as built urban areas for example, there are problems in both the number of visible satellites and in the environment relating to the signal propagation from the satellites. Signals may propagate in a direct path from satellite to receiver, be refracted around an obstruction, or through one or more reflections. Direct propagation and reflected propagation are called *line-of-sight* (LOS) and *non-line-of-*

sight (NLOS) propagation, respectively.

NLOS signals may cause two distinct types of errors: NLOS reception and multipath interference. In order to properly distinguish the two error sources the following terminology is adopted in this work; multipath interference refers to a situation in which both LOS and NLOS signals are received from a satellite whereas NLOS reception refers to a case of receiving only NLOS signals from a satellite. These two error sources differ in the ways they affect the positioning and in the means of mitigating said errors. [27] In a case of NLOS reception the delay caused by the increased distance between receiver and satellite will distort the observed ranging result. For multipath interference the errors will depend on the used positioning method and be more complex in general.

All obstructions big enough to block satellite signals may theoretically cause NLOS reception and multipath interference errors but the number of surrounding obstructions affects strongly the difficulty of the satellite navigation environment. Urban areas are often considered one of the most difficult environments for GNSS positioning [32]. Urban streets with buildings obstructing the sky-view on both sides of the street are referred to as urban canyons as an analogue to a real canyon. Urban canyons pose many challenges for GNSS positioning with limited sky-view and many obstructions capable of reflecting the signals.

In order to mitigate errors caused by the obstructions during signal propagation the *Intelligent Signal Masking for GNSS RTK* project—from here on ISMASK project—introduces a satellite masking technique to eliminate unwanted satellites from the RTK positioning computations. The satellite masking is a type of satellite selection designed to remove erroneous satellites from the GNSS processing. ISMASK is a research project carried out by SSF [38] and funded by European Space Research and Technology Centre (ESTEC) [2] which is a technology center operated by European Space Agency (ESA). This thesis is done as a part of the ISMASK project by one of the three SSF employees participating in the project.

GNSS positioning is studied in this work in the context of RTK positioning as the name of the ISMASK project implies. As a result of the positioning method, the research is done with high accuracy equipment and requirements of high precision positioning in mind. A similar system could very well be used in standard precision GNSS receivers but the results obtained in this study will apply only to RTK positioning.

A satellite masking system, referred to as ISMASK system, was developed and analyzed in the ISMASK project. The studied satellite masking method requires information of the obstruction environment in order to determine the obstructed satellites. In this project the information is provided by two methods—from a *digital surface model* (DSM) also known as 3D model of the surrounding environment and from a sky facing camera. Both of these methods are able to provide a map of the surrounding obstructions which can be translated to a mask of obstructions. The two methods, however, provide the obstruction information with different characteristics. GNSS positioning with the help of obstruction masking can be referred to as aided positioning as the existing GNSS positioning algorithms are aided by the additional satellite masking information.

There are theoretically two possibilities in the masking as the masking can be directed to either individual satellite signals or to the entire satellites. As all signals from a GNSS satellite are originating from the same source it is logical with this kind of masking system to mask the satellites instead of signals. In this study, the term signal masking is sometimes used but the masking is always directed to the satellite.

The most important object in this study is to assess whether or not this kind of satellite masking system is a beneficial addition to RTK positioning and the feasibility of such system. Advantageousness and feasibility are both tied to the performance of the aided system compared to the unaided one. In order to assess these effects, the masking system has to be evaluated using real observation data by comparison of the results obtained using aided and unaided systems. For the assessment purposes the ISMASK project included a test campaign in which the system was studied in action on urban environments.

The different ways of using the two masking data sources in mask generation is another important aspect in the study. The question is in its essence about how do the different mask generation methods differ in performance. The mask generation from sky-view images and 3D models can be performed by using either of the data sources as independent methods or by combining the two data sources. Finding an optimal data utilization method is a requirement for potential commercial usage of this technology.

The goal of the thesis is to solve a research problem which can be formulated as following research questions:

1. What kind of an effect does signal obstruction masking with 3D model and sky-view camera have on RTK positioning performance?

2. What is the optimal way to utilize the information provided by 3D model and sky-view camera?

The research questions can be answered if the system capable of satellite masking is tested with sufficient amount of real observation data and the effects of obstruction masking on the positioning solution are determined carefully from the data collected in the test campaign.

This study includes a presentation of the theoretical background of the developed system, description of the system and test campaign used to study the developed system, results of the test campaign, and analysis of the obtained results. This thesis presents the ISMASK project comprehensively from the developed system to tests, results, and analysis of the results.

2 Theoretical background

This section gives an overview of important theoretical concepts related to the developed satellite masking system. First part gives overview of GNSS positioning in general before the second part presenting theory of RTK positioning in particular. The third subsection gives important theoretical concepts of satellite filtering and masking including the used mask generation methods. Fourth section describes integrated GNSS-INS navigation systems which is an essential part of the test environment. The last part gives an overview of related studies presenting a cross-section of similar studies.

2.1 GNSS positioning

Global navigation satellite systems are satellite navigation systems able to provide users with a three dimensional positioning solution with global coverage. The term GNSS not only includes the well-known *Global Positioning System* (GPS) operated by United States but also other similar systems. GPS has become widely known and used due to being the first GNSS to reach operational capability. At the moment in addition to GPS, also the Russian equivalent *GLobal NAvigation Satellite System* (GLONASS) is fully operational. The Chinese BeiDou Navigation Satellite System has reached operational capability in the Asia-Pacific region and the global coverage is in development. The European Galileo satellite system has launched initial services with 18 satellites in orbit [16]. The full Galileo constellation of 30 satellites is currently in development with the target of full operational capability set for 2020 [16].

The first used satellite navigation system was the Transit system launched in 1960's by the United States. Transit utilized Doppler based positioning providing 2D position fixes with accuracy of tens of meters and low temporal availability. GPS—the first GNSS system—was declared initial operational capability in 1993 which made the Transit system obsolete and led to the discontinuation of the Transit navigation service in 1996. The Russian GLONASS was the second GNSS to reach full operational capability. [26, pp. 12–13]

Satellite based navigation systems—GPS especially—have become an integral part of our society with their ability to provide easy-to-use positioning in global coverage. Standalone GNSS positioning, for example GPS *standard positioning services* (SPS), provides accuracy on the meter level [26, p. 13]. For many usage purposes this is however not sufficient. GNSS positioning accuracy can be improved with different computational techniques such as precise point positioning and differential GNSS.

Composition of GNSS is often divided into the three segments [6]:

- space segment consisting of satellites in the orbit
- ground segment consisting of ground control and monitoring stations
- user segment consisting of all user equipment.

The greatest advantage of GNSS compared to any other positioning technology is the global coverage it provides. The global usability is obtained by placing

something between 24 to 30 satellites to the *medium Earth orbit* (MEO) and possibly additional satellites to the *geostationary* (GEO) or *geosynchronous* (GSO) orbits. The satellites orbiting the Earth form the space segment of the GNSS system. All GPS, GLONASS, and Galileo satellites are placed in MEO orbits with some differences in orbital parameters between the individual systems. The plan for BeiDou is to have a constellation of 27 MEO satellites with additional 5 GEO and 3 GSO satellites providing greater satellite availability at regions of interest (Asia-Pacific region). [22, pp. 108–151]

GNSS is a radio navigation system in which the satellites—space segment—transmit the navigation signal from space to the Earth in the microwave, or depending on the definition radio wave, frequencies. The navigation data is transmitted in the form of pseudorandom codes which are modulated into the carrier wave along with so-called navigation message. The pseudorandom codes allow calculation of the signal travel time which can be translated to the range between user receiver and the satellite due to known velocity of the electromagnetic radiation. [6, pp. 27–28]

In addition to the radio transmitters the satellites contain also receivers and antennas which are used to receive maintenance data from ground segment. The satellites are also equipped with at least three separate high-precision atomic clocks in order to guarantee reliable timing information which is essential to the GNSS usage due to very high velocity of the signals. [6, pp. 27–28] The satellites have also an electrical power system providing required energy using solar arrays and backup batteries. In order to keep the satellites in the proper orbits the satellites are also equipped with a propulsion system which allows small orbital maneuvers.

Ground segment, which is also often called control segment, controls the satellite system from the ground. A network of ground stations around the world monitor and track the satellites constantly and transmit the monitoring data to the control station, or stations depending on the GNSS system in question. The control station processes the collected data in order to compute satellite orbits and attitude, derive satellite clock corrections, and generate correction parameters and navigation data for the satellites. The obtained data is then transmitted to upload stations from where it is further on transmitted to the satellites. [6, p. 41]

The GNSS user segment consists of all user receivers and antennas. The user antenna receives the carrier wave signal transmitted from the satellite and converts the radiation to electrical signals for the receiver. Receiver then demodulates the signals with the help of the receiver clock as a time reference. The demodulated data is then processed to obtain range from satellite to receiver antenna. From four or more such ranges the receiver is able to determine its 3D position and time. [26, pp. 300–303]

All positioning systems induce some amount of errors to the provided positioning solutions. The risk level of these errors and other system vulnerabilities are quantified by different positioning performance parameters. Some of the performance parameters such as accuracy are familiar from everyday life. It is intuitive that most GNSS users require a certain level of positioning accuracy such as the robotic cars might require lane level accuracy in order to operate. Accuracy is the most widely used characterization for the performance of a positioning system. Another widely used

characterization for the performance of a positioning system is availability. Availability describes the portion of the time that the system is able to provide a solution with required accuracy to the user. [19, p. 8]

In aviation industry navigation performance is typically quantified in addition to accuracy and availability by also navigation systems' integrity and continuity. Integrity describes correctness of the solution provided by a navigation system and ability to detect when the system is unable to provide a solution with required level of reliability. For aviation purposes, integrity is a key characteristic of navigation systems, the goal being usually to minimize the integrity risk in navigation system. Continuity is the measure of how likely is an interruption to a navigation operation after said operation has been started. [19, p. 8]

Accuracy, availability, continuity, and integrity are however not the only measures of navigation system performance. In some contexts, a useful measure is the robustness of a navigation system. Robustness describes how well the navigation system is able to resist or withstand negative effects such as interference or multipath. Robustness against jamming and interference is of particular interest in military applications.

In this study the positioning performance of used positioning systems is assessed mainly from the viewpoints of accuracy and availability. Accuracy information may be assessed by comparing the positioning solutions to the ground truth reference provided by GNSS-INS (*inertial navigation system*) reference system. Availability can be assessed by simply measuring the portion of time the system will provide sufficiently accurate solutions; particular interest lies in the assessment of RTK fix availability. Integrity is not assessed in this study but it can be hypothesized that the successful filtering of faulty satellite signals will reduce the integrity risk. Continuity could be assessed by for example studying the distances without navigation solution as was done by Ji et al. [32]. However, due to similarity to availability, continuity is not addressed in this study, and the main focus will be in the accuracy and availability parameters.

Standard GNSS positioning accuracy in meter level is obtained by so-called code based positioning. Code based positioning utilizes pseudorandom codes—most typically GPS C/A code—modulated to the carrier signal. This standard positioning method relies on the fact that if we can determine signal receiving time (from the receiver clock), signal transmitting time (from the pseudorandom code) and the speed of the traveling signal (the speed of light) we can determine the range between satellite and receiver antennas. If we assume that there are no timing errors the equation for the range p between the receiving antenna and the satellite can be written as [26, pp. 303-307]

$$p = \tau c \tag{1}$$

where $\tau = t_r - t_t$ is the signal travel time, t_r and t_t are the signal receiving and transmitting times, respectively, and c is the speed of light.

From the known location of the transmitting satellite and one receiver-to-satellite range the receiver can be determined to lie anywhere on the sphere with radius of the given range surrounding the satellite. If we add a second receiver-to-satellite range observation, we can determine the position to be in the circle intersecting the

two spheres. By adding a third receiver-to-satellite range observation we constrain the receiver position to two possible points one of which can be discarded in most cases as it lies in an impossible position such as in space or inside the Earth. [26, pp. 303–307]

In the previous example the range observations were assumed to be error free but due to inaccuracies in used clocks the ranges are corrupted by timing errors. The GNSS satellites are equipped with atomic clocks that are able to provide high accuracy timing but the clocks in GNSS receivers are typically much lower grade timekeepers. In order to compensate the timing errors an additional range observation is required for the GNSS positioning. With the use of at least four satellites the receiver is able to determine its position, velocity, and time albeit various different error sources will limit the positioning accuracy to a level from meters to tens of meters.

Due to errors in the observed ranges the ranges are referred to as pseudoranges. The equation 1 for the pseudorange ρ between the satellite and the receiver becomes then [21, pp. 56–57]

$$\rho(t) = p(t, t-\tau) + c[\delta t_r(t) - \delta t_s(t-\tau)] + I(t) + T(t) + \epsilon(t) \tag{2}$$

where

$p(t, t-\tau)$ is the true range between the receiver at time t and the given satellite at time of signal transmission

$\delta t_r(t)$ is the receiver clock error at the time of signal reception

$\delta t_s(t-\tau)$ is the satellite clock error at the time of transmission

$I(t)$ is the signal delay caused by the Earth's ionosphere

$T(t)$ is the signal delay caused by the Earth's troposphere

$\epsilon(t)$ is the term for remaining errors which are not modeled such as propagation delays and receiver noise.

The components of the pseudoranges are all satellite specific except for the receiver clock error. Terms that are specific to individual satellites are from here on denoted with superscript to identify the corresponding satellite.

The term of interest in the equation 2 in the positioning sense is true range p. The range can be written in Cartesian coordinates as

$$p^k = \sqrt{(x_s^k - x_r)^2 + (y_s^k - y_r)^2 + (z_s^k - z_r)^2} \tag{3}$$

which can be written in vector form as

$$p^k = \left\| \mathbf{x}_s^k - \mathbf{x}_r \right\| \tag{4}$$

where $\mathbf{x}_s^k = \begin{bmatrix} x_s^k & y_s^k & z_s^k \end{bmatrix}^T$ denotes position of the satellite k, and $\mathbf{x}_r = \begin{bmatrix} x_r & y_r & z_r \end{bmatrix}^T$ denotes position of the receiver.

The user location coordinates can (usually) be solved when more than four pseudorange observations are available. It is important to note that the solved position will be in the same coordinate reference frame as the satellite positions. Often, especially in the case of GPS positioning, the used reference frame is World Geodetic System 84 (WGS-84). [21, pp. 57–58]

From the individual pseudorange measurements the standard way of estimating receiver location is by first linearizing the observations around an estimated position and receiver clock error. After the linearizing the solution will be iterated until the difference between estimated location and observations approach zero. The initial estimate of position can be set to for example center of the Earth if no better guess is available as the computation will often converge even with such a rough guess. [21, pp. 58–59]

By writing the equation 2 for satellite k in the vector form, grouping the satellite dependent error terms into a single error term e^k, and substituting $c\delta t_r$ with clock error bias term e_c we can write observed pseudorange that has been corrected as

$$\rho_c^k = \left\| \mathbf{x}_s^k - \mathbf{x}_r \right\| + e_c + e^k \,. \tag{5}$$

Receiver estimated pseudorange ρ_e^k can similarly be written as

$$\rho_e^k = \left\| \mathbf{x}_s^k - \mathbf{x}_e \right\| + e_{c,e} \tag{6}$$

where $\mathbf{x}_e$ denotes estimated receiver position while $e_{c,e}$ denotes estimated receiver clock error.

By calculating the difference $\delta\rho^k = \rho_c^k - \rho_e^k$ between corrected pseudorange and estimated pseudorange and applying the Taylor series expansion to the result we get [21, p. 59]

$$\delta\rho^k = -\boldsymbol{\ell}_{\text{unit}}^k \cdot \delta\mathbf{x} + \delta e_c + e^k \tag{7}$$

where

$\delta\mathbf{x} = \mathbf{x}_r - \mathbf{x}_e$ is the difference between the true receiver position and the estimated position of the receiver

$\delta e_c = e_c - e_{c,e}$ is the difference between the true receiver clock error and the estimated receiver clock error

e^k consists of all satellite specific error terms

$\boldsymbol{\ell}_{\text{unit}}^k$ is the satellite direction unit vector pointing from estimated receiver position to the satellite k.

The satellite direction vector is given by [21, p. 59]

$$\boldsymbol{\ell}_{\text{unit}}^k = \frac{\mathbf{x}_s^k - \mathbf{x}_e}{\left\| \mathbf{x}_s^k - \mathbf{x}_e \right\|} \,. \tag{8}$$

From the satellite direction vectors the measurement geometry matrix can be calculated as [21, p. 60]

$$\mathbf{G} = \begin{bmatrix} -\boldsymbol{\ell}^1_{\text{unit}} & 1 \\ -\boldsymbol{\ell}^2_{\text{unit}} & 1 \\ \vdots & \vdots \\ -\boldsymbol{\ell}^k_{\text{unit}} & 1 \end{bmatrix}. \tag{9}$$

With multiple pseudorange observations the pseudorange differences can be written in vector form as

$$\delta\boldsymbol{\rho} = \begin{bmatrix} \delta\rho^1 \\ \delta\rho^2 \\ \vdots \\ \delta\rho^k \end{bmatrix} = \mathbf{G} \begin{bmatrix} \delta\mathbf{x} \\ \delta e_c \end{bmatrix} + \mathbf{e} \tag{10}$$

where $\mathbf{e} = \begin{bmatrix} e^1 & e^2 & \dots & e^k \end{bmatrix}^T$.

A least-squares solution for the receiver position difference $\delta\mathbf{x}$ and the receiver clock error difference δe_c can be found by [21, pp. 60–62]

$$\begin{bmatrix} \delta\mathbf{x} \\ \delta e_c \end{bmatrix} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\delta\boldsymbol{\rho}\,. \tag{11}$$

New values for the estimated receiver position $\mathbf{x}_e$ and estimated receiver clock error $e_{c,e}$ can be calculated from the least-squares solutions for receiver position and receiver clock error differences as [21, p. 61]

$$\mathbf{x}_e^{i+1} = \mathbf{x}_e^i + \delta\mathbf{x} \tag{12}$$

$$e_{c,e}^{i+1} = \delta e_c \tag{13}$$

where superscript i indicates current iteration cycle. It should be noted that the clock error is calculated from the estimated position for each iteration and therefore after the convergence of solution for a given iteration the current clock bias δe_c will be the best estimate for updated clock bias [21, pp. 61].

The iteration is continued from the pseudorange estimation step (equation 6) until the applied correction approaches zero within a given tolerance. For more in-depth information of the code based positioning see for example books by Gleason and Gebre-Egziabher [20] or Groves [26].

A-priori quality of the positioning solution can be assessed from the available satellite geometry. An even distribution of the satellites in both azimuth and elevation provides the best quality solution, but in practice the satellite distribution in the sky varies with time. In unobstructed areas the satellite availability in azimuthal direction is roughly uniform over time but the availability in elevation varies a lot more, especially in higher latitudes [24, p. 208].

The standard metric used for the quality of satellite geometry is *dilution of precision* (DOP). DOP is used to quantify sensitivity of the errors in the GNSS

solution to the errors in pseudoranges [22, pp. 44]. When the satellite geometry moves from good to less-ideal—for example when the satellites move closer to each other—the change in the satellite geometry will "dilute" the precision of the obtained solution. In the previous example the DOP would increase from a "good" value to somewhat higher to indicate higher dilution of precision. [21, p. 63] For a good satellite geometry DOP can be considered to be around "a few" while DOP values larger than 5 are definitely non-ideal [6, p. 136]. The average DOP for an unobstructed observer using the nominal GPS constellation of 24 satellites at a latitude of 60° is 1.84 [26, p. 430].

DOP values are determined from the DOP matrix $\mathbf{H}$ which can be calculated from the geometry matrix as

$$\mathbf{H} = (\mathbf{G}^T\mathbf{G})^{-1} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} . \tag{14}$$

There are in fact multiple widely used DOP metrics for analyzing the effect of the satellite geometry. Previously mentioned DOP values referred to total DOP or geometrical DOP (GDOP) while other commonly used DOP metrics are position DOP (PDOP), horizontal DOP (HDOP), vertical DOP (VDOP), and time DOP (TDOP). The values are derived from the DOP matrix as [21, p. 63]

$$GDOP = \sqrt{h_{11} + h_{22} + h_{33} + h_{44}}\,,$$

$$PDOP = \sqrt{h_{11} + h_{22} + h_{33}}\,,$$

$$HDOP = \sqrt{h_{11} + h_{22}}\,,$$

$$VDOP = \sqrt{h_{33}}\,,$$

$$TDOP = \sqrt{h_{44}}\,.$$

Different DOP metrics refer to how the errors in pseudoranges do affect different parts of the GNSS solution. GDOP includes all individual DOP values while position and time DOPs are tailored to the variables of interest. PDOP is further on divided to vertical and horizontal components. Due to the limited spread of satellites in elevation direction as the Earth blocks elevations below 0° the VDOP is generally worse than HDOP despite VDOP including only one axis compared to the two axes in HDOP [26, pp. 424–429].

2.2 Real Time Kinematic positioning

Previously described positioning methods utilized pseudoranges derived from pseudo-random codes modulated to the carrier waves and broadcast from the GNSS satellites. The obtained accuracy of the code based positioning solutions is in the range from meters to tens of meters which is too large a deviation for many usage purposes.

By observing also the phase of the carrier wave in addition to the modulated pseudocodes and utilizing well-known processing methods the GNSS positioning can achieve accuracy from millimeters to decimeters [21, p. 68]. The higher accuracy obtained from carrier phase measurements is achieved due to significantly—more than 100 times—smaller wavelength of carrier waves compared to the effective wavelength of pseudocodes. Carrier phase measurements provide considerably less noisy observations than code based measurements.

There exist many different positioning methods which are used in carrier phase based positioning. In this thesis the interest is in real time kinematic (RTK) positioning. Different carrier phase based positioning methods have different characteristics and usage purposes. Real time kinematic—as the name implies—is intended for kinematic real time positioning. The term "kinematic" is used to distinguish kinematic and static positioning from each other as some carrier phase based positioning methods are intended to be used with only static receivers. For this thesis in addition to RTK, also *post-processed kinematic* (PPK) positioning method is of interest as the data is processed after the test campaign in post-processing. All GNSS processing is however done in the manner that simulates RTK positioning. PPK and RTK are very similar processing techniques and due to the fact that the practical tests are done in a way simulating RTK, the theory of PPK is not discussed further. Consequently, all results are applicable to RTK and PPK positioning both.

RTK is a relative positioning method in which two separate GNSS receivers are used. One of the receivers is set up on a known location as a base (or reference) receiver. The other receiver—rover receiver—is the one position of which is of interest. The base receiver transmits correction information along with the known location it is set up on, to the rover receiver. The rover receiver uses the base station data with its own observations to compute the vector from the base station to the rover receiver from which the position of the receiver can be computed. [6, pp. 166-169]

Carrier phase based positioning is able to achieve higher accuracies when compared to code based positioning due to the fact that carrier phase can be measured with the accuracy level of millimeters to centimeters while the code phase can be measured in the meter level accuracy. The higher accuracy of the obtained ranges provides higher accuracy for the positioning solution. [6, p. 105-106] The measurements of carrier phase are much less noisy than the code measurements which enables higher accuracy of the measurements and consequently more accurate positioning. The approximate noise of a traditional GPS receiver for C/A code measurements is 3 m compared to 2 mm of carrier phase measurements [54, p. 323].

While the phase of the incoming carrier wave can be measured accurately the range between the satellite and receiver—around $2 \cdot 10^7$ meters—consists of an unknown number of full wavelengths of approximately 20 centimeters and of the remaining phase difference [21, p. 67]. The number of full wavelengths cannot be directly measured causing ambiguity in the carrier phase based positioning. The process of determining the integer number of full wavelengths, known as integer ambiguities, is an essential requirement for obtaining a good quality RTK positioning solution. There are many different algorithms for the integer ambiguity resolution, but the methods are not discussed here. For more information on the ambiguity

resolution see for example books by Seeber [54], Gleason and Gebre-Egziabher [20], or Bhatta [6].

The resolution of integer ambiguities is also known as integer ambiguity fixing, emphasizing the idea that once integer ambiguity is resolved it stays fixed to the same integer value unless the carrier phase tracking lock is lost. Ambiguity resolution is an integral part of RTK positioning enabling highest accuracy measurements in the centimeter level. When the integer ambiguities are fixed the RTK solution is referred to as fixed solution whereas the solution in which the integer ambiguities are not known is called a float solution. The unknown nature of integer ambiguities leads to a possibility of falsely fixed solutions in which the ambiguity of at least one satellite is resolved to a wrong integer causing ranging error in the magnitude of carrier wavelengths. A bad quality fixed solution can therefore be even of worse quality than the corresponding float solutions [60].

Carrier phase is the fractional phase difference of the incoming carrier wave compared to the by the local oscillator in the receiver. When the receiver obtains a lock on the satellite it starts to track the phase of the carrier wave and count the number of full wavelengths. The initial integer ambiguity remains unknown but constant for as long as the satellite remains locked. If the satellite signal lock is lost for a duration no matter how short the integer ambiguity will reset and needs to be resolved again. [47, pp. 352–353]

Observations in carrier phase based positioning are of the so-called phase-range observable. The observable consists of the range due to fractional phase difference between the reference signal and satellite transmitted signal, the range due to number of ambiguous integer wavelengths, and of the range due to error terms similar to the pseudorange observations (see equation 2). The phase-range observable Φ can be written as [6, p. 109][47, p. 351–354]

$$\Phi = p^k + c(\delta t_r - \delta t_s^k) - I^k + T^k + \lambda N^k + \lambda(\phi_{s_0}^k - \phi_{r_0}) + \epsilon_{\text{eph}}^k + \epsilon_{\text{mp}}^k + \epsilon_n^k \qquad (15)$$

where

p^k is the true range between the receiver and the satellite k

δt_r is the receiver clock error

δt_s^k is the satellite clock error for satellite k

I^k is the signal delay caused by the Earth's ionosphere for satellite k

T^k is the signal delay caused by the Earth's troposphere for satellite k

λ is the wavelength of the carrier wave, for GPS L1 $\approx$ 19 cm and GPS L2 $\approx$ 24 cm

N^k is the integer ambiguity in cycles for the used signal

$\phi_{s_0}^k$ and ϕ_{r_0} are the initial fractional carrier phases for satellite k and the receiver, respectively

ϵ^k_{eph} is the error due to satellite ephemeris errors for satellite k

ϵ^k_{mp} is the NLOS and multipath interference error for satellite k

ϵ^k_n is the error due to measurement noise for satellite k.

In the equation 15 it is important to note that some of the variables depend on the used satellite and some depend on the used signals and the characteristics of carrier wave. The satellite dependent components are identified with superscript k as before but in addition to that carrier wavelength, integer ambiguity, ionospheric delay, carrier phases, NLOS and multipath errors, and measurement noise errors are specific to the used signal. The effect of ionosphere is also reversed in the phase-range equation (equation 15) compared to the pseudorange equation (equation 2). From the equations it can be observed that the ionosphere delays the code measurements but advances the carrier phase measurements.

The clock error terms δt_r and δt_s and the integer ambiguity term N in the equation 15 are linearly dependent which complicates the integer ambiguity resolution [54]. In order to mitigate these and other error terms such as ionospheric, tropospheric and ephemeris errors new observables are generated by differencing the observations. The most commonly used differences are

- between observations made with different receivers
- between observations made to different satellites
- between observations made at different epochs.

Single difference observable is (usually) formed by differencing observations made with different receivers. The single difference observable allows mitigation of the ionospheric, tropospheric and ephemeris errors and eliminates the satellite clock error term. The degree of the error mitigation depends however on the baseline length between the two receivers. The negative side of the single difference observable is that it increases the noise level by the factor of $\sqrt{2}$ when compared to the noise level of the original observables. [47, pp. 354–356]

By denoting the phase-range observable between satellite k and receiver r as Φ^k_r the single difference observable between base receiver b and rover receiver r can be written as[47, p. 355]

$$\Phi^k_{b,r} = \Phi^k_b - \Phi^k_r \,. \tag{16}$$

Double difference (DD) is the most commonly used of the derived observables. Double difference could theoretically refer to any two times differenced observable but it is often used to refer specifically to the observable obtained by differencing two single difference observables between different satellites. The double differenced observable is in the essence a between receiver - between satellite difference combination. [47, pp. 354–356]

The double difference observable allows the mitigation of both satellite and clock errors due to the differences being generated at same time. Double difference also

allows reduction of the ionospheric, tropospheric and ephemeris errors with the magnitude of reduction depending on the length of the baseline. The disadvantage with DD observable is that the measurement noise is increased by the factor of 2 when compared to the noise level of the original observables. [6]

Double difference observable between satellites k and l for receivers b and r can be written as

$$\Phi_{b,r}^{k,l} = \Phi_{b,r}^{k} - \Phi_{b,r}^{l} \,. \tag{17}$$

The third commonly used difference is triple difference which is calculated by taking difference of two double difference observables made at two separate epochs. The biggest advantage of the triple difference is that it allows elimination of the integer ambiguity present in the other observables in addition to the clock errors already eliminated in double differences. [64, p. 295] Triple differences are often used to form approximate solutions, and in the detection and removal of cycle slips [54, pp. 261–262].

The triple difference observable between epochs t_1 and t_2 can be written as

$$\Phi_{b,r}^{k,l}(t_2, t_1) = \Phi_{b,r}^{k,l}(t_2) - \Phi_{b,r}^{k,l}(t_1) \,. \tag{18}$$

New observables can also be created by forming linear combinations of the individual frequencies. In theory there are unlimited number of possible linear combinations to be generated from the observations but in practice only limited number differences are commonly used. The formed linear combinations are often used to mitigate the effect of ionosphere in the measurements, and to aid ambiguity resolution. The beneficial effects of the combinations come at the cost of increased noise in the generated observables. [54, pp. 258–265].

Commonly used linear combinations include wide-lane, narrow-lane, and ionospheric-free observables. The wide-lane observable is obtained by $L_{\mathrm{WL}} = \phi_1 - \phi_2$ while the narrow-lane observable is obtained by $L_{\mathrm{NL}} = \phi_1 + \phi_2$, where ϕ_1 and ϕ_2 correspond to double differenced observables in L1 and L2 frequencies, respectively. [47, p. 358] The wide-lane observable has wavelength of 86.2 cm making ambiguity detection much easier with the observable, the downside being growth in the noise level to six times as large. The narrow-lane observable has wavelength of only 10.7 cm posing difficulties to the ambiguity resolution, but providing also the lowest noise level of all the linear combinations. By taking mean of the wide-lane and narrow-lane observables we get the ionospheric-free linear combination $L_{\mathrm{iono}} = \frac{L_{\mathrm{WL}}+L_{\mathrm{NL}}}{2}$. The ionospheric components in the wide-lane and narrow-lane observables are of equal magnitude but with different signs and hence eliminating the ionospheric component from the derived observable. [54, pp. 258–265] The ionospheric-free observable is usually used for long baselines where the ionospheric effect grows too strong to be modeled with other methods.

In order to solve ambiguity and bias terms in the equation 15 there are two distinct approaches. Either eliminating the parameters by taking differences described above, or trying to estimate the parameters. The parameter estimation technique can utilize additional observations such as ionospheric delay measurements or use of an extended

adjustment model to estimate the effect. In the parameter estimation method, the non-differenced observables are used in the processing. [54, pp. 265–266]

Real time kinematic positioning can be implemented with the use of double differences between base and rover receivers or by using the parameter estimation method. In RTK the base receiver broadcasts corrections continuously to the rover receiver by using for example a radio link or an Internet connection. The corrections can be represented in either observation space or state space. In observation space the pseudorange and carrier phase corrections are represented as lump sum corrections while the state space representation achieved with the parameter estimation method can be used to generate corrections for individual error sources [66]. After the rover receiver has received the correction data from the base station receiver it will attempt to compute the baseline vector using a method suitable for the provided correction data. [6, pp. 166–169][54, pp. 336–338]

The integer ambiguities are resolved in RTK positioning with so-called *on the fly ambiguity resolution* (OTF-AR). On the fly ambiguity resolution allows determination of the integer ambiguities while the rover receiver is in motion. OTF-AR is a requirement for *truly* kinematic positioning as the ambiguities will need to be fixed while moving and re-initialized after cycle slips. [54, p. 294]

After the receiver has generated double differences and resolved ambiguities, it will have to use the resolved integer ambiguities to process the ambiguous double differenced observables to unambiguous double differenced range observables. From the unambiguous range observables, a baseline vector between base and rover receivers can be computed. Finally, the baseline vector can be used to compute the receiver position with the use of the known location of the base station. [47, p. 358]

There are a few things worth of noting in the computation of the RTK solution. Firstly, the accuracy of the resulting solution is dependent on the accuracy of the base station coordinates [6, p. 168]. Secondly a requirement for higher accuracy applications such as surveying is that dual (or triple) frequency receivers are required for the ambiguity resolution [54, p. 337]. It should be also noted that unless integer ambiguities can be resolved or resolved with required reliability the RTK processing software may keep the solution as an ambiguous float solution.

Typical accuracy for RTK positioning with sufficient number of satellites is approximately from 0.02 to 0.2 meters for baselines shorter than 20 km [48, p. 195]. Typical horizontal accuracy for a RTK position is 1-2 cm + 1 ppm and 1.5-5 cm + 2 ppm in the vertical measurements [6, p. 342]. Ppm refers to parts per million, i.e. millimeter for each baseline kilometer in the estimates above.

The processing phases of a GNSS receiver are divided here to five different blocks presented in figure 1. At first, GNSS antenna receives signals from GNSS satellites which feeds the obtained signals to the RF front end. The RF (radio frequency) front end processes the RF data into digitized signals and feeds the data onwards to baseband processing block. The baseband processing block is responsible of Doppler removal, code generation, correlation and integration processes. The integrated correlator outputs are then input to ranging processor block which generates observables such as pseudo- and phase-ranges along with navigation data. These observables are then used in application processing to compute required information,

most commonly a PVT solution. [26, pp. 349][20]

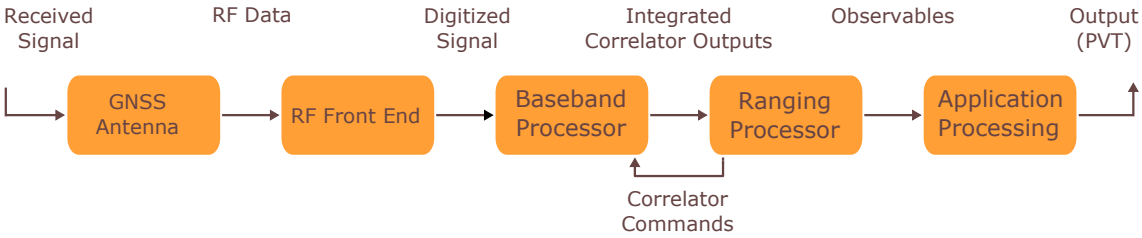

Figure 1: Processing blocks of a conventional GNSS receiver

The antenna is the part responsible of receiving GNSS signals and passing them on to RF front end. Multi-frequency receivers typically contain separate front ends for each frequency band [26, pp. 352–353]. Main functions of a RF front end include filtering, amplification and in most cases down-conversion of GNSS signals to intermediate frequency (IF). The RF front end then feeds the IF signal to the analog-to-digital converter from which the sampled digital signal is input to the next block. [22, pp. 199–204]

The following block, baseband processing, is traditionally done with dedicated receiver hardware but nowadays software-defined receivers allow also baseband processing with general-purpose processors [26, pp. 356]. Baseband processing and ranging processor are done in parallel channels, one channel being responsible of a single signal from an individual single satellite. The implementation presented here is a typical one, presented also in for example an introductory navigation book by Groves [26].

Baseband processing begins with a Doppler wipe-off step in which the digitized signal is split to in-phase (I) and quadrature (Q) streams which are offset by 90°. The signal is split to I and Q streams by multiplying the digitized satellite signal samples by in-phase and quadrature samples of receiver-generated carrier. The receiver carrier signal is generated from the current Doppler estimate and numerically controlled oscillator (NCO) data. The obtained I and Q signal samples are then transmitted to correlator where they are multiplied by early, prompt, and late reference codes. Prompt code is aligned with the received signal while early and late codes are separated from prompt code by a code-phase offset which is traditionally set to between 0.05 and 1 code chips—chip referring to an effective bit of a pseudorandom code [7, p.24]. Correlator outputs are then integrated over a predetermined integration interval. [26, pp. 355–359] The presented baseband processing scheme applies only to signals modulated with binary phase-shift keying (BPSK) which is the traditional modulation used in GPS satellites.

The ranging processor block is often classified as a part of the baseband processor but in the implementation presented here a division between the two phases is included. The logic in this classification is that baseband processor and ranging processor blocks mark the division between hardware and software sections of the GNSS processing in the majority of currently used receivers [26]. The ranging processor takes as input the integrated correlator outputs to generate GNSS observables. The ranging processor is usually implemented in separate channels for each signal as is the case with the

baseband processing phase.

The first step in the ranging processor block is a coarse synchronization of the code phases between reference and observed signals. The coarse synchronization step is referred to as signal acquisition. In order to determine signal transmission time, the code phase of reference signal must be matched to the observed signal. Due to relative motion of the transmitting satellite and user receiver the carrier frequency will be Doppler shifted and hence unknown. Consequently, also the frequency of reference and observed signals must be estimated and matched in addition to the matching of code phases. The acquisition requires a two dimensional search for code delay and Doppler frequency. [26, pp. 367–372]

After reference and observed signals are synchronized to sufficient accuracy—typically 0.5 chips—a fine synchronization of signals must be performed and maintained. The fine synchronization is performed in signal tracking loops. Without signal tracking the residual Doppler of observed signal would cause the obtained signal alignment to drift out and eventually the received signal would vanish below the noise level. Maintaining the fine synchronization with the reference code timing is called code tracking while maintaining the frequency between the observed carrier and the reference signal is called carrier tracking, both of which are needed in GNSS positioning. [22, p. 213]

Code tracking is usually implemented in a delay lock loop (DLL) which uses accumulated early, late, and prompt correlator outputs of I and Q to measure code tracking error. The code tracking error is then used to correct the estimate of the code phase. Estimated code phase is used not only in pseudorange calculation but also fed-back to baseband processing to control the NCO. [22, p. 213–217]

Carrier tracking may be implemented as carrier phase tracking or carrier frequency tracking, or both. Carrier phase tracking eases demodulation of navigation data and is a requirement for carrier phase based positioning methods, including RTK. Carrier frequency tracking provides better performance with poor quality signals and in environments with high–dynamics maneuvering. A common implementation method is to use frequency tracking as an intermediate phase before carrier phase tracking to take advantage of strengths of both tracking methods. [26, pp. 377–389]

Carrier phase tracking is usually implemented as phase lock loop (PLL) while the typical frequency tracking implementation is frequency lock loop (FLL). Carrier tracking, unlike code tracking, utilizes only prompt correlator outputs from baseband processing. The output of PLL is the estimate of carrier phase error with possibly also an estimate of carrier frequency error. The output of FLL is the estimate of carrier frequency error. The estimated carrier phase can be used to determine accumulated delta range, while from the estimated carrier frequency the pseudorange rate can be computed. These observables along with the pseudoranges obtained from code tracking can be used in the application processing block to compute RTK solutions. [26, pp. 377–389]

2.3 Satellite filtering and masking

In order to obtain a GNSS solution the receiver antenna must be able to receive signals from the satellites. GNSS signals can propagate straight from satellites to the receiver antenna or be reflected from one or more obstructions delaying the received signal. Reflected signals are delayed compared to signals propagating in a straight path from satellite to receiver resulting in range observations inconsistent with the geometric range between satellite and receiver. [27]

The basic concept of NLOS reception, multipath interference and LOS signals is presented in figure 2. In the figure LOS signals are drawn in green, NLOS and blocked signals with red, and NLOS signal from the satellite where a LOS-signal also exists in blue. Therefore, blue signal will mark the existence of a multipath interference caused by reception of both LOS and LOS signals from the same satellite. In the given scenario satellite SV1 would cause NLOS reception error, SV3 would cause said multipath interference error, and satellites SV2 and SV4 would be propagation-wise good quality signals.

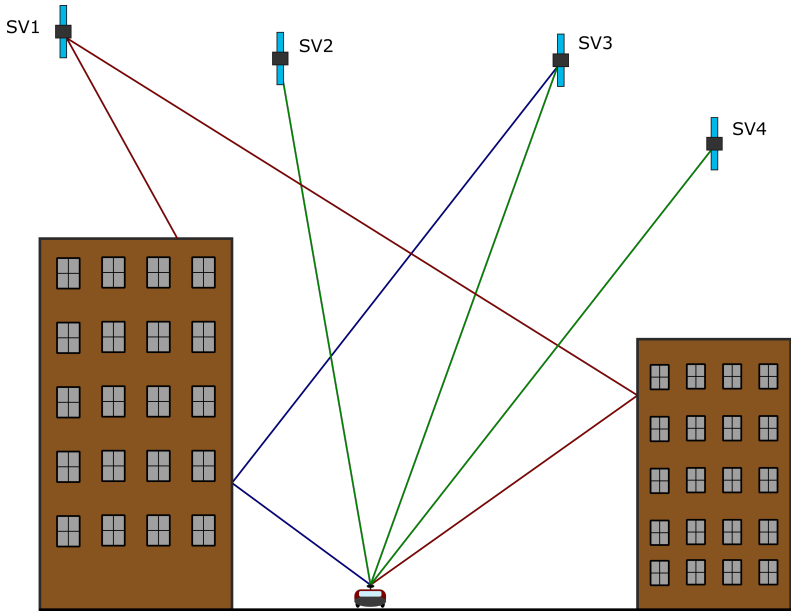

Figure 2: LOS-only reception, NLOS reception and multipath interference

The effects of multipath interference and NLOS reception are often referred to as multipath error, there is however a distinction between the two errors and in the ways the errors can be mitigated [27]. From here on in this work only the exact terms *multipath interference* and *NLOS reception* are used to refer the two phenomena and their respective errors.

Multipath interference will negatively affect the receiver performance in both code and phase tracking. For RTK and other carrier phase based positioning methods the carrier phase tracking errors due to the multipath interference will be limited to a quarter of the carrier wavelength (approximately 5 cm for GPS L1) [27][41]. Whereas the NLOS reception errors are not bounded, potentially causing much larger errors [46].

NLOS reception and multipath interference are major error sources for the GNSS positioning, especially in environments with a lot of obstructions such as urban canyons [22, p. 264]. Urban areas with tall buildings are extremely prone to cause errors due to signal propagation delay and at worst unavailability of positioning solution. Signal tracking loss due to obstructions is not uncommon in the environments with tall obstructions and narrow streets. Tall buildings obstructing the sky view may lead to outages in PVT solution ranging from a few seconds to minutes depending on the satellite navigation environment [18, p. 159].

The increased number of satellites available due to the development of the Galileo and BeiDou constellations guarantee better satellite availability even in difficult environments. Additional satellites do not however automatically increase the GNSS positioning accuracy due to potential errors in the measurements. Even one satellite with NLOS reception error can induce large errors to the positioning solution [58].

Obstructed urban environments—urban canyons especially—are challenging environments for GNSS positioning not only due to the small number of available signals but also due to poor obstruction geometry. The obstruction geometry in a long urban canyon might be good in the along-road direction, but in the cross-road direction the buildings obstruct the sky view resulting in weak satellite geometry [63]. The poor signal geometry in addition to NLOS reception and multipath interference makes urban canyons extremely difficult satellite navigation environments.

GNSS receiver may implement *receiver autonomous integrity monitoring* (RAIM) algorithms in order to attempt detection and possible exclusion of faulty signals. RAIM algorithms determine the integrity of the GNSS solution by performing a consistency check amongst the observations. There are various different types and ways to implement a RAIM algorithm, but the general idea is to check the consistency of observations, using for example pseudorange or least-squares residuals. The satellites with non-consistent observations may then be excluded from the GNSS processing with the aim being improved quality of the resulting solution. The steps described above are referred to as *fault detection and exclusion* (FDE). After FDE the traditional RAIM algorithms compute protection levels for the solution. Protection levels correspond to the radius of a circle surrounding the obtained position which contains the true position with required integrity risk. [11]

Traditional RAIM algorithms are not designed particularly for mitigation of NLOS or multipath interference errors. Especially in urban environments where the portion of satellites with LOS only reception might be considerably lower compared to less obstructed environments the conventional RAIM algorithms lack sufficient effectivity [33]. Some RAIM like satellite selection algorithms have been studied with the improvement of urban usage in mind [27].

Satellite masking refers here to a concept of dividing available GNSS satellites to subsets using some available information source. The subsets may be for example satellites to be accepted and rejected from GNSS processing. The rejected satellites can be then filtered out from the computations, or instead of excluding the satellites their effect can be also reduced by weighting the unwanted satellites with lower weight than other satellites. The filtering refers here to the process of totally excluding the data of filtered satellites from the processing algorithms. In satellite masking

systems such as the ISMASK system presented here the filtering is a simple process of applying the satellite mask to the set of available satellites. The FDE satellite exclusion in RAIM algorithms is essentially also a filtering process.

Satellite filtering is not the only way to deal with multipath interference and NLOS reception. Different ways to mitigate multipath interference exist with different benefits and disadvantages [8]. The observations of NLOS satellites may be corrected by adjusting the satellite-receiver range if information of the surrounding environment such as a 3D model is available, instead of just excluding the satellite from processing. Pseudorange multipath interference errors may also be corrected by using statistical data [15], while the carrier phase observations have been successfully corrected for multipath interference with information known about the received signals albeit the method works only for statistical measurements [49].

Methods for mitigating the NLOS reception and multipath interference may be implemented in many parts of the GNSS positioning process. The methods can be categorized to antenna-based, receiver-based and to post-receiver techniques [26]. Antenna based mitigation can be used to attenuate reflected signals. However, some of the antenna based techniques only work in mitigating multipath interference but have no effect to NLOS signals, while some of the techniques are not practical due to size and economic constraints they induce [27].

Receiver-based mitigation methods may also be used to reduce the errors related to multipath interference but they will not allow mitigation of the NLOS reception errors [27]. Many of the receiver based methods are done in the baseband processing phase. Commonly used such methods include use of narrow correlator [14], Double-Delta correlator [10], and *multipath estimating delay lock loop* (MEDLL) [61]. All of these methods have proven to provide better results than the conventional delay lock loops [14][61][10].

The receivers used in the ISMASK project—and consequently in the field tests presented in chapters 3 and 4—use receiver manufacturer Septentrio's *a posteriori multipath estimation* technique (APME). APME method attempts to mitigate multipath interference by using a special multipath estimator module in a setup with narrow correlator combined with an extra very late correlator. The multipath estimator module estimates the magnitude of code tracking error due to multipath interference using accumulated correlator outputs from the prompt and extra correlators. [57] The APME implementation in receivers used in this work is able to estimate multipath interference error for carrier phase tracking in addition to the code tracking implementation presented by Sleewaegen and Boon [57]. The details of APME with carrier tracking—referred to as APME+ by Septentrio—are however not published information.

Post-receiver based mitigation techniques are applied before or during the GNSS processing algorithms. The post-receiver methods typically use GNSS observations and possibly additional data in the mitigation process. The post-receiver based filtering may utilize for example *signal-to-noise ratios* (SNR) or stochastic modeling [37][49]. Many of the post-receiver techniques without external data do not, however, affect NLOS reception. Post-receiver techniques include also consistency checking methods operating in similar logic to RAIM algorithms [27]. These techniques may

also be improved by integrating external information such as 3D model data to the existing algorithms [29].

The concept of satellite masking is an integral part of GNSS positioning especially when more accurate positioning is needed. Many GNSS receivers and processing software implement a simple cutoff angle to filter out low elevation satellites from positioning computations. The signals from low elevation satellites have to travel a longer way through the Earth's atmosphere than the signals from the satellites at higher elevations and therefore are likely to have larger errors due to atmospheric effects. Low elevation satellites suffer also from higher level of multipath interference and from lower carrier to noise density ($\frac{C}{N_0}$). Typically used cutoff angles are around 10 degrees. [22, p. 155] By excluding the low elevation satellites the GNSS processing is able to reduce the magnitude of propagation related error sources including ionospheric errors.

The satellite masking used in this work takes the masking concept further by creating an obstructions mask of the environment surrounding the receiver antenna. This obstruction mask is then used similarly to elevation mask to filter out unwanted satellites from the positioning computations. The satellite masking is performed before the position computation and as such the masking method presented here can be classified as an external data utilizing post-receiver NLOS mitigation technique. The obstruction masks are generated from sky-view images and from a digital surface model. The 3D model may be used to generate also a mask of the multipath interference, which can then be used in conjunction with the NLOS reception mask to filter out both sets of potentially erroneous satellites.

Example masks are presented in figures 3 and 4, generated from sky-view image and digital surface model, respectively. The example masks are generated for the same epoch. The masks follow normal sky plot presentation convention with 0, 30 and 60–degree elevation circles displayed with blue and 10–degree elevation mask displayed in yellow. The plot axes are oriented as North-Up, East-Right.

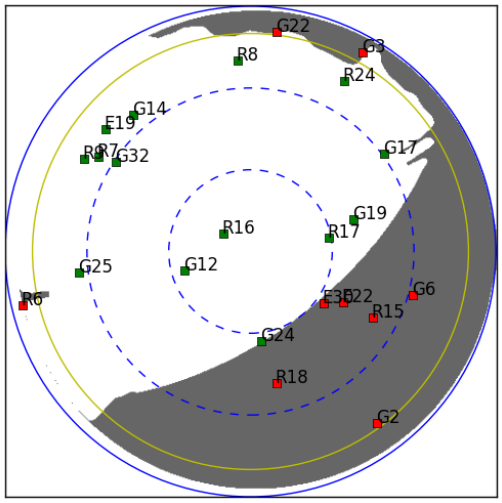

Figure 3: An example signal mask generated from sky image

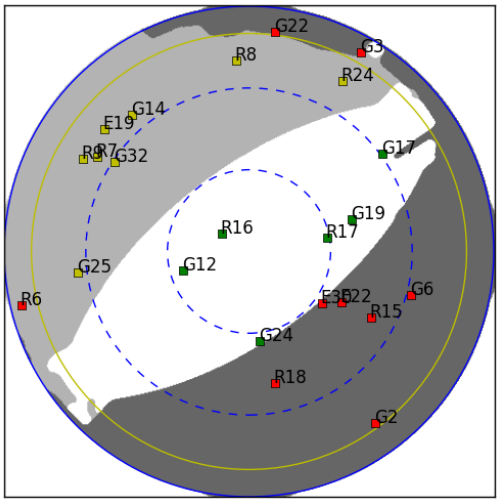

Figure 4: An example signal mask generated from DSM

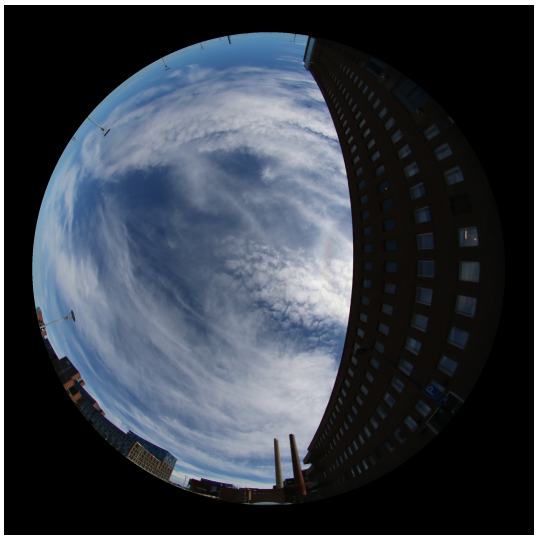

Figure 5: A source image for the example mask

Areas with obstructions are presented in the masks with dark gray while non-obstructed sky areas are displayed as white. The DSM mask has in addition to obstructed and unobstructed areas also light gray areas which depict unobstructed areas which have potential for multipath interference.

The masks also display visible satellites over the horizon with squares colored according to the masking status—green for good visibility, yellow for moderate visibility, and red for blocked. The moderate visibility category is used to define satellites which may or may not be used in the processing depending on the number and geometry of good visibility satellites. A moderate quality indicator is used in the surface model based mask for the satellites with potential multipath interference. The source image used in the image based mask generation is presented in figure 5. The orientation of the source image is not transformed to the navigation frame so the upward direction in the image corresponds to the vehicle forward direction instead of pointing North.

In theory, if all of the NLOS satellites are filtered out from the positioning computations, the NLOS reception error will be eliminated from the solution. There will however be left the part that results from multipath interference of the satellites that are received via both LOS and NLOS propagation. Due to imperfections in the satellite masking the effect will not naturally be 100% but if the masking is able to exclude even most of the NLOS satellites the error will likely be mitigated considerably. The idea in satellite masking is that mitigating an error source as large as the NLOS reception will have positive impact on the performance of RTK positioning, especially in urban canyons.

The ability to detect multipath interference areas is a feature in the DSM based masking method which will be presented later. The information of the potential multipath interference areas can be used to disregard the satellites that may contain multipath interference errors from the positioning computations. Due to the small magnitude of the said errors there is, however, a doubt whether or not it might be beneficial to filter out the masked satellites, as it will lead to degradation of the

satellite geometry. The multipath interference is also at least partially compensated by the receivers utilizing Septentrio's proprietary multipath mitigation technique which should improve the quality of observations effected by multipath interference.

Mitigating the NLOS reception and multipath interference errors—especially the NLOS reception—has potential to improve the RTK positioning performance by eliminating large error sources hampering the positioning performance.

Digital surface models—commonly referred to as 3D models or city models—contain georeferenced three dimensional models of the terrain and depending on the model also possibly buildings, structures and trees [9]. The information contained in the 3D models combined with known locations of GNSS satellites can be utilized to identify satellite visibility conditions from a known location.

The 3D city models are all simplifications of the real-world object geometries containing only static objects such as buildings and large structures and potentially some other objects. The magnitude of simplification is quantified by *level of detail* (LOD) of a given 3D model. The levels of detail defined in the CityGML standard [13]—a standard defining information model for 3D representation of cities and landscape—run from LOD0 to LOD4. LOD0 is the most generalized model while LOD4 is the most accurate description of the real-world environment. LOD0 models contain regional to landscape level data, LOD1 contains city to region scale information, LOD2 is for city to city district level models, LOD3 is used for city district to exterior architectural model level data, while the most accurate LOD4 level information contains also interior architectural model level data [13].

The most useful LOD levels for satellite masking are levels from LOD0 to LOD2 as the extra details from higher LOD levels do not provide much extra benefit at the cost of increased data size. LOD0 level data may also be unusable due to being too generalized in details. In this project the used models are LOD1 consisting of terrain and building information. The buildings in the used DSM are generalized to rectangular boxes without dedicated roof structures.

Digital surface models are nowadays freely available in some areas and commercially available on many other areas, but due to high costs of the model generation there are many locations in which the models are not available. However, 3D city models are often available in urban and dense urban areas. The models used in this project are provided free of charge by the city of Helsinki in the Helsinki Region Infoshare service [34].

The signal mask is generated from the DSM model by the means of ray tracing which is a method of choice also in various other similar studies [62][42][29]. Ray tracing originated from image generation needs in computer graphics but it can also be used to model propagation of radio waves, such as GNSS signals [31]. Ray tracing allows the path of GNSS signals to be estimated with great accuracy at the cost of computational load. The computers are nowadays, however, able to handle the computational load caused by the ray tracing, even in real time solutions.

The downside of the mask generation from digital surface models is that the approximate location of the receiver must be known prior to the mask computation. One solution for this is to calculate a dead reckoning solution from the latest PVT solution. A dead reckoning solution can be computed as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_k t \tag{19}$$

where $\mathbf{x}_k$ is the receiver location at epoch k, $\mathbf{v}_k$ is the receiver velocity at epoch k and t is the time between epochs.

In the algorithm implemented in the ISMASK project, the mask generation begins by initializing a mask with a certain number of individual pixels. In the tests conducted in this work the used mask size was set to 500×500 pixels. Each of these pixels are mapped to a direction vector originating from the estimated position using an equidistant fisheye projection in which angles of incidence are translated linearly into radial distances, thereby preserving angle information needed in the ray-tracing [53].

The mask pixel coordinates used here abide by the following convention: x-axis is oriented from mask left to mask right, y-axis is oriented from mask bottom to upwards, and origin being at the mask center. The direction (unit) vector $\mathbf{x}$ is calculated from mask pixel coordinates by

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{x_{\text{px}}}{R} \sin(\phi) \\ \frac{y_{\text{px}}}{R} \sin(\phi) \\ \cos(\phi) \end{bmatrix} \tag{20}$$

where

$\mathbf{x}$ contains x, y, and z coordinates of the direction unit vector

x_{px} is the pixel x-coordinate

y_{px} is the pixel y-coordinate

R is the radial distance from pixel to image center $R = \sqrt{x_{\text{px}}^2 + y_{\text{px}}^2}$

ϕ is the angle between the real world point and the optical axis defined by $\phi = \frac{R}{\sqrt{n_{\text{px}}/\pi}}$
where $\sqrt{n_{\text{px}}/\pi}$ corresponds to the focal length of the projection.

A ray is then traced from the DSM model in the direction of the computed direction vector. If the ray does not intersect the DSM model the pixel is assumed to correspond to the visible sky. In a case that the ray intersects the model then the pixel corresponds to an obstruction. Areas with potential multipath can be computed by tracing a reflected ray from points of intersection. The direction in which the reflected ray does not intersect the 3D model can then be classified as possible multipath interference regions. Similar tracing of reflected rays can be done for any number of possible intersections but due to attenuation of the reflected signals the additional benefit might not outweigh the additional processing load caused by extensive ray tracing. Multipath interference tracing is thereby limited in this work to only first level intersections.

By processing each mask pixel in the described manner the computation results in an obstruction mask consisting of visible, blocked and potential multipath areas.

The obtained mask can then be used to filter unwanted satellites from the GNSS processing.

The method of generating signal mask from 3D model described above is however not the only possible way. The source of 3D model information does not have to be a digital surface model, but can be for example a 3D *geographic information system* (3D GIS) [17] or a 3D building model [45]. Different 3D information sources have different characteristic advantages and disadvantages. The use of DSM provides a complete description of the receiver surroundings—constrained by model LOD—providing a large amount of information, some of which will be unnecessary. The disadvantage of DSM is the size of data. The data used in this work required from 1 to 10 MB for 1 square kilometer sized model blocks depending on the complexity of the environment. Assuming mean size of 5 MB per block the entire land area of the city of Helsinki would require 1 GB storage.

Usage of signal masking in large areas requires a large amount of 3D data for signal mask generation, causing eventually problems with storage capacity. While the storage problem can be overcome by storing the 3D model data on the server and downloading the required data on demand, another option is to pre-compute signal masks and generate a grid of signal masks and use the pre-computed signal masks instead of computing a new mask for each epoch. [23] The pre-computed mask method degrades however mask accuracy, as very small grid spacing would cause enormous amount of processing and also result in vast amounts of data. The DSM storage size could be improved also by optimizing the 3D data for satellite masking usage and by using data compression methods.

An alternative—or complementary—method to obstruction mask generation to DSM derived masking is to use sky-facing images of GNSS antenna surroundings as data source for obstruction masking. Provided the sky area is detected from the image and camera orientation is known, the image can be used to generate a mask of satellite visibility obstructions and visible sky.

The strengths of image derived masking compared to digital surface models is that it does not require an estimated position, it is up to date, contains also non-static obstructions such as large vehicles, and that it does not require additional data. The masking requires however an attitude solution for the rover, daylight for imaging, and perhaps the most demanding requirement is that it requires additional hardware in the form of a camera in addition to possible additional hardware for attitude determination.

In this work a standard *digital single-lens reflex* (DSLR) camera with circular fisheye lens is used. The field of view of the used lens is 180° meaning that if pointing approximately to zenith the whole sky above 10° elevation mask will be captured in the image. In addition to the camera image, also attitude information of the image is required for satellite masking. In this work the camera orientation is determined from the GNSS-INS reference system which provides attitude solutions at 200 Hz rate. Alternative options for attitude determination would include multi-antenna GNSS setups and dedicated inertial navigation systems.

The camera zenith pointing condition is fulfilled with required accuracy on flat surfaces. While roving around the city it is common that due to topography the

camera will not be pointing directly in the zenith. While driving for example uphill the camera will face slightly downwards of zenith causing parts of the mask in the direction of the vehicle movement to not appear on the image. The areas left out of the image are classified as undefined pixels and the corresponding satellites are included in the processing unless some other information, i.e. 3D model based mask, is available.

Image based masking has several requirements in addition to the camera orientation. First the image needs to be divided to pixels corresponding to sky and to pixels corresponding to obstructions. The process of determining which pixels are to be masked and which are not is referred to as sky detection. After sky detection, the masking information needs to be transformed from image pixel coordinate system to mask coordinate system. The process of mapping each image pixel coordinates to mask pixel coordinates—or equivalently to azimuth and elevation—requires calibration of the used camera system. In fact, for any metric information to be obtained from 2D images, the camera system needs to be calibrated. [52]

Camera calibration refers here to a calibration of the camera intrinsic parameters which are required in order to map image pixels to directions. The imaging function—mapping each pixel to a direction—is described by coefficients of a Taylor series expansion which are estimated in the calibration process. Camera calibration used in this work is done with *OCamCalib: Omnidirectional Camera Calibration Toolbox for Matlab* presented by Scaramuzza et al. [52] and extended by Scaramuzza et al. [50] and Scaramuzza [51]. The toolbox works in Matlab and allows calibration of cameras with fisheye lenses.

The calibration is performed by using images of a planar checkerboard pattern. The toolbox includes automatic corner point extraction for the pattern which automates the calibration process by a great deal. The calibration used in this work has in total five different sets of outputs. The first output set are polynomial coefficients of the image mapping function, mapping each pixel to a direction vector on the unit sphere centered at the single effective view point of the camera system. The second output are polynomial coefficients of the inverse mapping function mapping each unit sphere direction to a pair of image pixel coordinates. The third output are image center coordinates. The last output are parameters of an affine transformation which consists of corrections to errors in the digitizing process and small errors in alignment of the camera axes [52].

The camera calibration has to be performed only once for the camera and the calibration values can then be saved and used later. In addition to intrinsic calibration, also extrinsic calibration could be performed to obtain precise camera orientation parameters but in this work the extrinsic calibration was not performed. Instead of the extrinsic calibration, the camera was installed in the test campaign in a way that the orientation is known with the required accuracy level.

Image processing starts with the standard fisheye JPEG image, an example of which can be seen in figure 3. There are various different methods to perform the sky detection. The different sky detection techniques utilize different aspects of the source images. Arguably simplest of the algorithms use only RGB color values of each pixel to determine if it matches the presumed color of the sky [30]. Information

such as color and texture or mathematical morphology can also be used [4]. Sky detection algorithms often simplify the original image to reduce computing load and take advantage of homogeneous regions in the images by performing a segmentation of some sort to the images. Segmentation methods such as k-means clustering have been used on satellite detection [4]. In this study a superpixel clustering method is adapted to sky detection providing sufficient computational load along with robust superpixel generation sky detection.

Superpixels are pixel groups in which perceptually meaningful regions of the image are used to replace individual pixels [1]. These groups preserve among other features also the boundary of sky regions. There are various different superpixeling methods with different advantages and disadvantages. In this study, the *simple linear iterative clustering* (SLIC) superpixeling method is adapted. SLIC is an adaptation of k-means clustering with mainly improvement in computational performance and control of resulting superpixel characteristics [1]. For more on SLIC and other superpixeling methods see for example Achanta et al. [1].

All sky detection related image processing operations are performed in CIELAB color space in which the traditional RGB color components are transformed to L, a, and b components. The L component corresponds to lightness while the a and b components are two color opponent dimensions [55, pp. 30–32]. The used SLIC algorithm is implemented in CIELAB color space by default—or more correctly the algorithm transforms RGB images to CIELAB color space for segmentation.

The general logic behind the implemented sky detection algorithm follows the algorithm presented by Kostolansky [35] which is also performed in CIELAB color space. Consequently, the algorithm is performed in CIELAB color space. The general logic of algorithm follows the following logic:

1. Generate superpixels with SLIC algorithm
2. Find out mean color value for each superpixel
3. Calculate statistics using mean colors of superpixel
4. Calculate threshold for superpixel clustering from statistics
5. Merge adjacent superpixels to larger clusters using the calculated threshold
6. Determine which merged cluster corresponds to sky area.

After the image is divided to a number of superpixels these superpixels are then clustered to larger superpixel clusters using a simple method iterating through all superpixels and checking whether or not the color distance between neighboring superpixels is below calculated threshold. The color distance d between superpixels 1 and 2 is given by

$$d = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2} \tag{21}$$

where L_i, a_i, and b_i are the L, a, and b color component values of superpixel i, respectively.

The threshold required for superpixel merging requires an initial guess of potential sky area. In this work the assumed sky region is a small square area in the image center with the length of the side of the square being 0.3 times the radius of the circular fisheye image area. The threshold T for superpixel clustering is determined from color distances between neighboring superpixels by

$$T = (d_{\text{sky}} + (d_{\text{all}} - d_{\text{sky}})\, C_1)\, C_2 \tag{22}$$

where

d_{sky} is the mean color distance between neighboring superpixels in the assumed sky region

d_{all} is the mean color distance between neighboring superpixels in entire image but disregarding 30% of the lowest distances

C_1 and C_2 are constants defined empirically using the values from Kostolansky [35] as a baseline, in this work the constants C_1 and C_2 were set to 0.1 and 2.3, respectively.

The clustering yields a clustered image in which nearby superpixels are merged together based on their color similarity in CIELAB color space. The clustering threshold is designed to separate sky from obstructions regardless of sky color in all kinds of conditions be the weather either sunny, cloudy, or partially cloudy. The sky cluster is determined to be the cluster with most pixels inside the assumed sky region.

The sky will not, however, likely be contained in a single cluster due to inhomogeneous composition of the sky. For instance, the Sun will often burn out the image resulting in large color distance to the rest of the sky. In order to compensate for this, all clusters enclosed inside the borders of the sky region are merged to the sky cluster. This newly formed sky cluster will then be classified as open sky and the pixels outside of this cluster will be classified as obstructions. The resulting image which contains pixels classified as sky areas and obstructions is referred to as masked image.

A flow diagram of the sky detection algorithm is presented in figure 6. The captured image is processed with the algorithm described here to a masked image in which image pixels are replaced by pixels containing masking status of each pixel, i.e. the information if pixel is part of sky cluster or not. In this way the status is set to visible for all pixels in sky cluster and to blocked for all other pixels.

After the masked image is generated, the obstruction mask generation is performed by transforming the masking information to a mask structure with a different coordinate system. The transformation goes by converting each mask pixel to a direction vector originating from unknown user location. The direction vector is then transformed to image pixel coordinates using the known attitude of the camera and obtained calibration parameters, namely the inverse mapping function. The obstruction status of the mask pixel will then be set to the status of masked image pixel before proceeding to the next mask pixel.

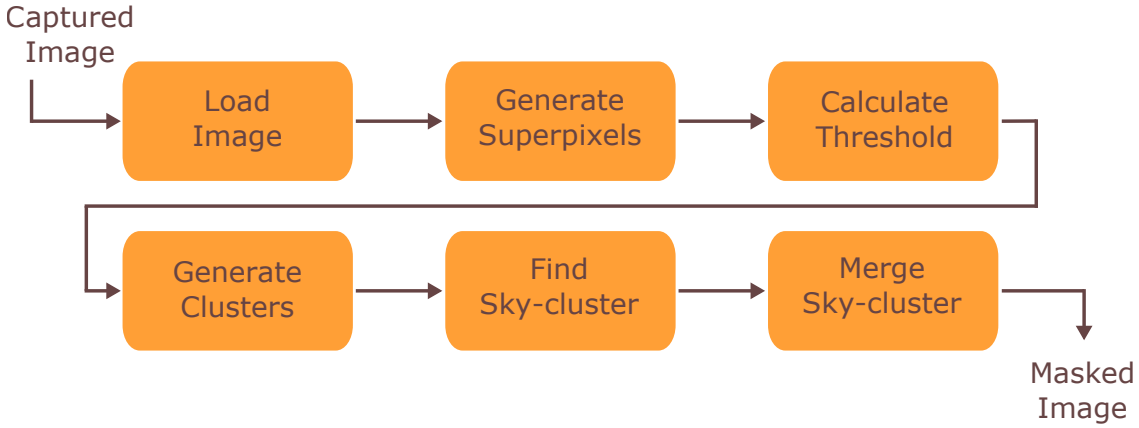

Figure 6: Flowchart of sky detection algorithm

2.4 GNSS-INS reference system

The performance of the masking system used in the ISMASK project will be assessed using a reference solution obtained with a reference navigation system. The idea in reference system is to provide presumably higher level accuracy in the used environments than the standard RTK accuracy of the GNSS receivers used. In the test campaign a Trimble MX-2 mobile mapping system was used as the reference system.

Inertial navigation systems provide an independent navigation solution by means of dead reckoning from accelerometer and gyroscope measurements. Inertial navigation systems are self-contained systems able to function without relying on external data after the system has been initialized. The ability to provide an independent navigation solution has made inertial navigation an integral part of aviation and modern warfare as planes and tactical missiles for example require reliable navigation data [65]. An important characterization of inertial navigation systems is that the errors in solutions are time-correlated and grow in time due to accumulation of errors in successive inertial measurements [26, p. 7].

An INS consists of an inertial measurement unit and a computer system that computes a navigation solution from the measurements and initial states. An *inertial measurement unit* (IMU) incorporates (typically) three accelerometer and three gyroscope sensors mounted in an orthogonal triad measuring acceleration and rotation of the navigating body.

Navigation with inertial measurement systems relies on a well-known relationship between position, velocity, and linear acceleration [22, pp. 10–11]

$$\mathbf{v}(t) = \mathbf{v}(t_0) + \int_{t_0}^{t} \mathbf{a}(s)ds\,, \tag{23}$$

$$\mathbf{x}(t) = \mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{v}(s)ds\,. \tag{24}$$

From the equations 23 and 24 we can see that position can be determined if the initial position and velocity along with subsequent linear accelerations are known.

Initial position and velocity are initial states which are needed from an external source as they cannot be measured with inertial measurement systems. IMU accelerometers are used to measure the linear accelerations required in equation 23. In order to use the linear accelerations in a real implementation they would have to be measured in a non-topocentric reference system which IMU is not as the linear accelerations are measured relative to the navigating body. To overcome this problem, the linear acceleration measurements are accompanied with attitude measurements. The orientation of the navigating body and therefore also the measured linear accelerations are measured with gyroscopes in IMUs. By combining these two measurements a dead reckoning solution may be calculated.

As we see from equations 23 and 24 the accelerations—measurements from an accelerometer—must be integrated twice. Accelerometers are subject to noise which causes errors in the measurements which grow as function of t^2 due to the integrations. In addition to the accelerometers, also gyroscopes suffer from noisy measurements. Gyroscope errors also grow in time but as a function of t^3. [18, p. 158]. The growth of errors leads to accumulation of said errors and consequently to degradation of the obtained position. Instability of the navigation solution is a major drawback in inertial navigation.

The performance of inertial navigation systems depends on the quality of used inertial sensors. The extremely high cost of high-end inertial sensors is one of the largest drawbacks of the technology. An estimate from 2013 valued inertial navigation systems which are able to provide a meaningful navigation solution without external aiding for more than few minutes after initialization to approximately 80 000 € [26, p. 559].

The strengths and weaknesses in GNSS and INS systems complement each other which provides a major advantage for an integrated navigation system combining GNSS and INS sensors. The integrated navigation system consisting of GNSS and INS is able to provide a better navigation solution than either of the subsystems alone. [22, p. 30]. The performance of a GNSS system can increase in all normally used metrics: accuracy, integrity, continuity and availability [19, p. 9]. Integration also increases the bandwidth of the navigation solution compared to the GNSS solution and provides the GNSS user with orientation information [18, p. 162]. GNSS can also provide the inertial navigation system with initial states which the system is unable to measure. The increase in performance when compared to a GNSS only receiver is the most obvious in situations where the sky is temporarily not visible as self-contained INS can provide a navigation solution for a limited time during a GNSS outage.

There are different ways to integrate GNSS and INS systems offering different combinations of strengths and weaknesses of the combined systems. The selection of integration method depends on the used hardware and navigation requirements. Integration architectures may differ in many aspects. Different implementation schemes are divided often into three different types: loosely, tightly and ultra-tightly coupled integration [26, pp. 560–562].

The most fundamental difference between different integration architectures is the type of data (measurements) used from the GNSS receiver for navigation solution

computation. Tighter architectures take advantage of INS data at an earlier stage of GNSS processing or deeper inside the receiver architecture. Loosely coupled integration combines a GNSS PVT solution with inertial measurements providing virtually two solutions: a GNSS solution and a combined solution. Tightly coupled integration combines GNSS observations—code and carrier phase measurements—with INS acceleration and angular velocity measurements. Ultratightly coupled integration is performed at a level where INS is practically included in the GNSS receiver, as the INS measurements are incorporated at signal tracking level into the system. [26, pp. 560–562]

The tightly coupled integration—such as the one used in Trimble MX-2—provides improved accuracy when compared to the accuracy of the individual systems and also when compared to the loosely coupled integration. The integration also improves continuity, availability, and robustness of the system compared to standalone GNSS due to being able to withstand outages. A tightly integrated system has also a benefit of being able to aid inertial measurements with less than 4 GNSS satellites which would be required for a GNSS solution in loosely coupled integration hence providing greater availability in obstructed areas. [18, p. 162–165] The ability to perform in low visibility conditions such as urban canyons or other heavily obstructed areas makes the tight integration architecture well suitable for use cases in such environments as is the case in this study.

The navigation system in Trimble MX-2 is able to provide a reliable reference solution for used RTK solutions as the system is able to perform in difficult satellite navigation environments than the RTK-only positioning. The difference in solution quality is significant especially in heavily obstructed environments such as the urban canyons on dense urban areas.

Despite better performance in obstructed environments the integrated GNSS-INS navigation system requires good satellite visibility from time to time. If the GNSS solution is lost the system might require a stop in a good visibility location to improve the solution quality. The length of such outages without acceptable GNSS solution is dependent on the quality of used INS. The Trimble MX-2 system should be able to handle GNSS outages lasting a few minutes but longer periods without sufficient satellite visibility are bound to cause problems if the system is wanted to provide a solution with accuracy higher than the nominal RTK accuracy. These considerations are taken into account in planning and execution of the test campaign.

2.5 Related studies

The large magnitude of NLOS and multipath interference errors in combination with the increase in the number of available satellites due to development of the Galileo and BeiDou constellations has led to a great deal of research on satellite selection and filtering algorithms. Many studies are focused on urban and dense urban areas as the multipath interference and NLOS reception are dominant error sources in such environments. Using aiding information such as 3D models has become popular in recent years [29], while also other aiding methods like fisheye cameras have been studied [58].

Most of the research on NLOS reception and multipath interference mitigation using methodologies similar to the ones used in this work is done in the context of code based positioning. Code and carrier phase based positioning methods differ in the ways the NLOS reception and multipath mitigation affect positioning, such as multipath interference errors in carrier tracking being bounded by a quarter of used wavelength. The accuracy levels of the positioning methods are also vastly different resulting in different magnitudes of the errors in question. However, the way in which cameras and 3D models are used is essentially identical between studies focusing on either of the positioning modes.

3D models have been utilized in the determination of the GNSS multipath environment for almost 10 years. Prediction of the satellite availability from 3D models with successful implementation was presented by Bradbury et al. [9]. The possibilities of 3D models in GNSS performance evaluation in terms of availability, integrity and precision have then been assessed in more detail [62]. The studies have shown that 3D models can reliably be used to evaluate visibility of GNSS satellites.

Different kinds of 3D models, including DSMs, along with camera based systems have been utilized in NLOS satellite masking. 3D GIS based information has been used to detect and filter NLOS satellites providing improvement in positioning accuracy, especially when used with an integrated GNSS-INS navigation system [17].

NLOS detection and exclusion using 3D building models combined with road maps has been shown to improve performance of code based positioning in the context of vehicular positioning. The positioning performance was improved in terms of both integrity and accuracy. The NLOS filtering had a positive impact on the positioning solutions in different navigation setups consisting of GPS-only, GPS and GLONASS, and GPS-INS integrations. [44][45] It has been found that accuracy of combined code based GNSS positioning and vehicle odometry system does not improve with 3D model based NLOS satellite masking compared to SNR based satellite filtering unless the positioning system was augmented by map aiding in addition to the odometry [46]. The map aiding method used in the study used geometric information of the road stored in GIS to estimate vehicle motion.

Filtering NLOS satellites with the obstruction masks derived from images have also been studied. A masking with a far infrared camera in RTK positioning was found to be able to improve solution quality by mitigating large (blunder) position errors. The border between sky and obstructions is more distinctly visible on infrared images than on visible light images easing the sky detection. The system was able to improve RTK positioning accuracy also in static conditions in addition to kinematic conditions. [40]

A GNSS integrity monitoring solution for an integrated GNSS-INS navigation system utilizing a visible light fisheye video camera and vector delay/frequency lock loop [36] is also proposed [56]. The robustness of the system is based on the concept of the camera allowing exclusion of the NLOS satellites and vector delay/frequency lock loop enabling mitigation of multipath interference. The combined effect enhances position with respect to two different error sources.

The camera based NLOS satellite filtering method has also been studied in RTK positioning usage [58]. In the study the (visible light) camera images were used in

conjunction with odometer data in order to perform visual odometry from which the camera orientation was obtained. The sky detection utilized also the known location of assumed LOS satellites in order to compute a likely sky area contrary to the constant region used in this work. The masking method was found to improve RTK fix ratio from 58.9 % to 61.2 % and to increase positioning accuracy.

Ray tracing on 3D models enable accurate computation of the whole path of GNSS signals. The NLOS errors arise from the delay caused by extra distance the signal has traversed due to indirect path between transmitter and receiver. By tracing the path from receiver to obstruction and further on to the transmitting satellite, the distance error in satellite-receiver range can be computed and NLOS reception error compensated. This kind of method in code based GNSS positioning has been demonstrated and found to improve accuracy and reduce integrity risk in positioning solutions [42][5]. The difference in range measurement accuracy between code- and phase based positioning means that the results from studies on code based positioning are not applicable to phase based positioning such as RTK.

A different approach to the GNSS navigation in urban canyons is so-called *shadow matching* presented first by Groves [25]. Shadow matching aims to improve GNSS positioning accuracy in urban canyons by utilizing the invisible satellites as an additional information source. The logic in shadow matching is that we can use a 3D model to predict which satellites should be visible on different locations and compare this information with the observed satellites to determine where the observed and predicted visibilities match. Shadow matching is particularly suitable for improving cross-street accuracy of code based GNSS positioning in urban canyons [23][28][63]. It has not, however, been studied in the context of carrier phase based positioning.

3D models can also be used to aid GNSS positioning in different ways. One possibility is so-called *height aiding* in which the height information obtained from a DSM or height database is used to generate a virtual range measurement from the center of the Earth to the location defined by estimated horizontal position and height from the external source. Height aiding has been found to improve observation geometry and as a result also accuracy of position solutions in code based positioning. [3][27]

3D models and camera data can be used also as a direct positioning source augmenting the other navigation sensors when needed. A system combining code based GPS, INS, camera, and 3D GIS has been presented and tested by Cappelle et al. [12]. The system tries to match the camera image to a position in a 3D GIS to provide correction data for INS when the GNSS signals are unavailable. The combined system aims to improve availability of position solutions in the absence of GNSS signals instead of just improving the existing GNSS solution.

Despite the research carried out on NLOS and multipath interference mitigation in recent years, the feasibility and performance of external data aided RTK positioning is still somewhat an open question. The following chapters present methodology and results obtained in the ISMASK project with the goal being answering the open questions.

3 Research material and methods

The satellite obstruction masking system developed in the ISMASK project—referred to as ISMASK system—was thoroughly evaluated in the ISMASK project. The system can be divided into two parts: ISMASK software and ISMASK test bench. The next chapter gives an overview of how the ISMASK system functions in the context of the two parts. After the system overview the test campaign in which the ISMASK system was evaluated is presented. Finally, the methodologies used in the analysis are described.

3.1 ISMASK system

As stated above the ISMASK system studied in this thesis consists of software and hardware parts. The whole system was developed during the project using both *commercial-of-the-self* (COTS) and custom solutions developed for this particular project.

A high level overview of the processing steps in the ISMASK software is presented in figure 7. The software begins by initializing GNSS processing software and loading required data such as 3D models. The RTK processing is performed in a loop in which the software calculates first a RTK positioning solution for an epoch. After processing the first epoch the software generates an obstruction mask for the next epoch using selected data source for the mask. The generated obstruction mask is then applied to the set of available satellites in next epoch filtering out unwanted satellites. The loop continues by processing the next epoch using a new, possibly filtered, set of satellites. The ISMASK software can consequentially, be dived into three separate sections: obstruction mask calculation, signal filtering, and RTK processing. The three sections are each represented by a box in the figure.

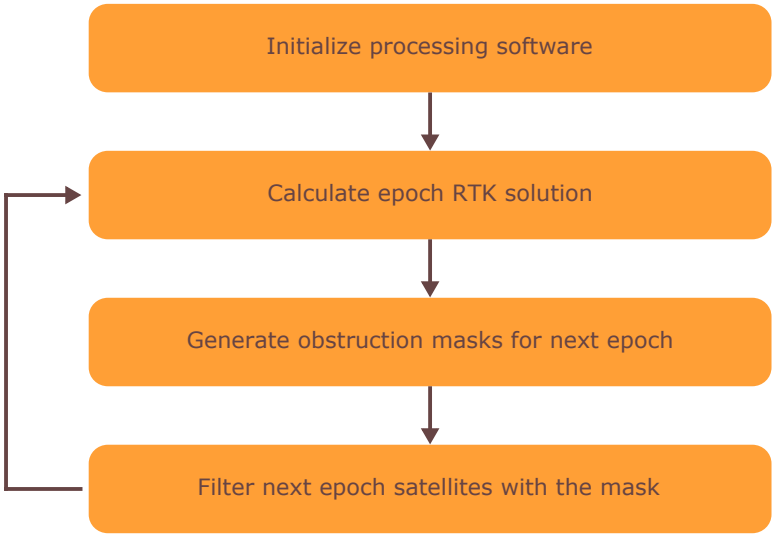

Figure 7: ISMASK software processing overview

The ISMASK software uses the obstruction masking data either from only the

images, only 3D models, or from both sources, to process the RTK GNSS solutions. The software takes the GNSS and masking data as input, generates the signal masks, filters out the satellites, and processes the GNSS solutions with filtered satellites outputting the newly generated solution.

The software side consists of readily available parts and parts developed for the purposes of the ISMASK project. The main COTS portion of the software is the RTK processing software package. The initial choice for RTK processing software was the software developed by receiver manufacturer Septentrio, due to easy integration with used receivers and flexible processing due to provided *application programming interface* (API). The Septentrio processing software is called Septentrio *Post Processing Software Development Kit* (PPSDK), for more information about the software see Septentrio website [43].

At the time of the project, Septentrio PPSDK was, however, unable to use Galileo satellites in RTK positioning. Assessment of signal masking in the context of Galileo enabled positioning was an important aspect in the project—due to ESA being funder of the project—causing a problem for RTK processing with only Septentrio PPSDK. In order to accommodate Galileo satellites to the project, RTKLIB was selected as a second GNSS processing software package. RTKLIB is an open source RTK processing software available on the Internet [59].

As open-source software RTKLIB has a lot of differences compared to the proprietary Septentrio PPSDK. The open nature of RTKLIB means that the used algorithms can be inspected by anyone and are mostly documented, while the algorithms used by the Septentrio software are disclosed only very briefly and in little detail. It can also be assumed that the Septentrio software is optimized for the use with Septentrio hardware. Septentrio PPSDK can also take advantage of extra data recorded by the Septentrio receivers such as multipath estimates recorded by APME+ which presumably are used to allow more advanced positioning algorithms than the ones implemented in more general software such as RTKLIB.

One of the biggest differences in the two GNSS processing software is that RTKLIB is unable to reliably resolve integer ambiguities for GLONASS satellites due to GLONASS inter-frequency biases, resulting from the use of different receivers for rover and base station. Septentrio PPSDK uses their own proprietary techniques to calibrate the inter-frequency biases and therefore is able to resolve ambiguities also for GLONASS satellites. The resulting scenario is that Septentrio PPSDK can use GPS and GLONASS satellites for all solutions and GPS, GLONASS, and Galileo for non-RTK solutions, while RTKLIB can use all three constellations but does not fix GLONASS integer ambiguities. As a result, considerably fewer satellites in RTKLIB RTK mode can be fixed, which leads to inferior performance in terms of the quality of fixed solutions compared to Septentrio PPSDK.

Most of the processing options that could be configured by the user such as elevation masks were set to identical values for both processing applications in order to have as similar characteristics as possible between the solutions of the two software packages.

The developed ISMASK software generates signal masks from the sky-view image and DSM data as described in section 2.3 and applies these masks to the set of

available signals. The processing software uses the set of good quality signals to compute RTK solutions. This software structure enables comparing the effect of the filtered solution to the unfiltered solutions processed otherwise in a similar manner as the masking and filtering phase may be configured to use different masking methods or no masking at all.

The obstruction mask can be generated from images or DSM, 3D model based generation providing two different methods for mask generation; with and without masking of multipath interference areas. In addition to these three methods there is also a fourth possibility for masking as the image and DSM based masks can be combined. A fifth testing mode for the system is to run without mask calculation phase at all and use all available satellites for input to GNSS processing software. In this work the five alternative test modes are evaluated with the goal being identification of the optimal method for mask generation.

It can be hypothesized that the multipath interference masking may not be beneficial to the positioning performance, especially in obstructed environments such as urban and dense urban areas, due to the small magnitude of the multipath interference errors and pre-existing (receiver based) multipath mitigation. The masking will, besides mitigate multipath errors, also degrade the observation geometry in already obstructed environments. DSM based mask generation will be used in with and without multipath interference masking modes in order to obtain information on the effect of such masking modes.

Obstruction masks can be visualized as sky-plot like images in which every pixel is assigned a visibility value based on the data obtained from the mask source. In the ISMASK software four different values were used for the pixels: good, moderate, blocked, and undefined visibility. In image based masking the values can be good, blocked, or undefined. Good and blocked being self-explanatory and undefined used for pixels that don't appear on the image due to camera not pointing zenith. DSM based masks consist of the same three values, but in multipath interference masking mode the multipath interference areas are mapped with moderate quality. The mask can be set to undefined for example if the estimated position falls inside of a building.

Moderate quality pixels are blocked from the PVT computations unless the a-priori quality estimators of the PVT solution don't meet the required level. A-priori quality of the solution is estimated with PDOP and the number of available satellites. The limit for exclusion of moderate quality satellites was set to DOP larger than three and number of satellites to at least 5 plus one for each used constellation for which at least one satellite being is tracked. The goal of the quality estimators is to avoid masking satellites that might contribute to NLOS or multipath interference errors but at the same time are vital to the observation geometry or otherwise needed in PVT computation. The set levels are relatively high which is intended to reduce over-aggressive satellite filtering due to imprecisions in the masking process.

Image and DSM based masks can be combined to form a hybrid mask by utilizing data from both sources. DSM masks are, however, used only in the non-multipath interference masking mode to avoid complications with moderate quality pixels. The mask combination in the ISMASK software is implemented at the mask level, meaning that the obstruction masks are generated in the same manner as when processing

with a single data source and the generated masks are then combined by comparing the visibility values of each mask pixel and setting the value to blocked if it is blocked on both masks. If a pixel is blocked only in one mode and visible on another mode, the pixel is set to moderate quality in the combined mask. Moderate quality areas function otherwise similarly to the multipath interference areas, meaning that they are blocked from the PVT computations unless the a-priori quality estimators of the PVT solution don't meet the required level. Pixels that are undefined in one mask, due to for example camera not pointing directly at zenith or missing 3D model, are set to the value of the other mask. This combination technique is less strict than an alternative implementation method which would block pixels that are blocked on either of the masks. The less-strict version was deemed to be more suitable because too strict filtering will likely cause problems in environments with a small number of available satellites.

After the mask is generated using the chosen method, an additional smoothing is applied to the mask in the form of a median filter. This filtering is intended to reduce the mask noise. This noise arises from the source data and is most evident in multipath interference masking with 3D models. Digital surface models are a simplification of real world object geometry causing dubious results in ray tracing from more complex surfaces, especially from curved surfaces such as smokestacks. The use of a median filter improves the quality of the generated masks by removing noise in many such cases. The median filter compares values of neighboring mask pixels and sets the value of a pixel to the median of the chosen number of nearby pixels. In the ISMASK project the median filter radius was set to 5 pixels. The radius of said median filter was determined empirically by comparing the results of RTK processing using different radius values and choosing the one which provides the best results.

After mask generation the signal masks are applied to the observation data. The signal filtering process starts with checking the visibility value of signal mask in the direction of each visible satellite. Each satellite in blocked and moderate areas is then filtered out, either by sending an exclude satellite command to the PPSDK software or by removing the corresponding observation from the RINEX observation file in RTKLIB processing. The difference in filtering between the two software packages comes from the different capabilities of the two packages. If the a-priori quality estimators of the PVT solution are on a suitable level, the RTK processing starts with either of the two software. Should the quality estimation fail to provide sufficient accuracy, the moderate quality satellites are included and another quality check is performed. In a case that the quality estimation is still not on required level, all satellites are included in the computation.

Obstruction masking, especially from sky-view images, has a potential risk in too aggressive filtering when passing an object momentarily from close-by. For example, light posts or trees next to the road might cause filtering to exclude satellites behind the obstructions for a single epoch. The single epoch filtering might unnecessarily require operations such as ambiguity resolution to start from nothing and therefore degrade solution quality. In the tests performed in this work, an additional step was included before the satellite exclusion to prevent too aggressive filtering in such

instances by requiring each satellite to be obstructed in at least two continuous epochs before excluding them from the RTK processing. The two epoch requirement was a result of small scale empirical tests in which the two epochs was determined to be an optimal value.

After the ISMASK software has determined the best set of satellites the processing moves to RTK processing in either RTKLIB or Septentrio PPSDK software. The two software packages are used for the same function of RTK processing but there are differences in the processing algorithms. The nature of the software is also different as PPSDK is proprietary software sold by Septentrio whereas RTKLIB is free open source software developed by volunteers.

In order to assess how does the signal obstruction masking implemented in the ISMASK software affect the RTK positioning performance, the system was tested in a test campaign. The test campaign required for a test bench to be built. The ISMASK test bench contains the hardware equipment necessary for collecting the test data in the ISMASK project. The majority of test equipment was installed to a car which is used to collect the test data. In addition to the equipment installed to a car, also an additional GNSS receiver acting as a RTK base station belongs to the test bench.

The ISMASK test bench consists of following equipment:

- base station GNSS receiver (Septentrio PolaRx-5)
- base station GNSS antenna (Septentrio PolaNt-x MF)
- rover GNSS receiver (Septentrio AsteRx-U HDC)
- rover GNSS antenna (Septentrio PolaNt-x MF)
- camera system (Canon EOS-70D with Sigma 4.5mm f/2.8 EX DC Circular Fisheye lens)
- reference navigation system (Trimble MX-2)
- laptop controlling the measurement system
- power source for the laptop and the GNSS receiver.

The test bench installed to the rover vehicle roof rack is presented in figure 8. The camera is connected to the operating laptop with a USB cable and the ISMASK antenna is connected to the ISMASK receiver with a standard coaxial antenna cable. The parts of the reference navigation system that are installed to the vehicle roof are connected to power source and operating laptop with corresponding cables.

The rover equipment was installed in a car roof rack as depicted in figure 8. The base of the installation was a long beam, which is a part of the reference navigation system. In addition to the reference system equipment, also the ISMASK camera and rover antenna were attached to the same beam allowing installation of all equipment to a parallel line in the vehicle movement direction.

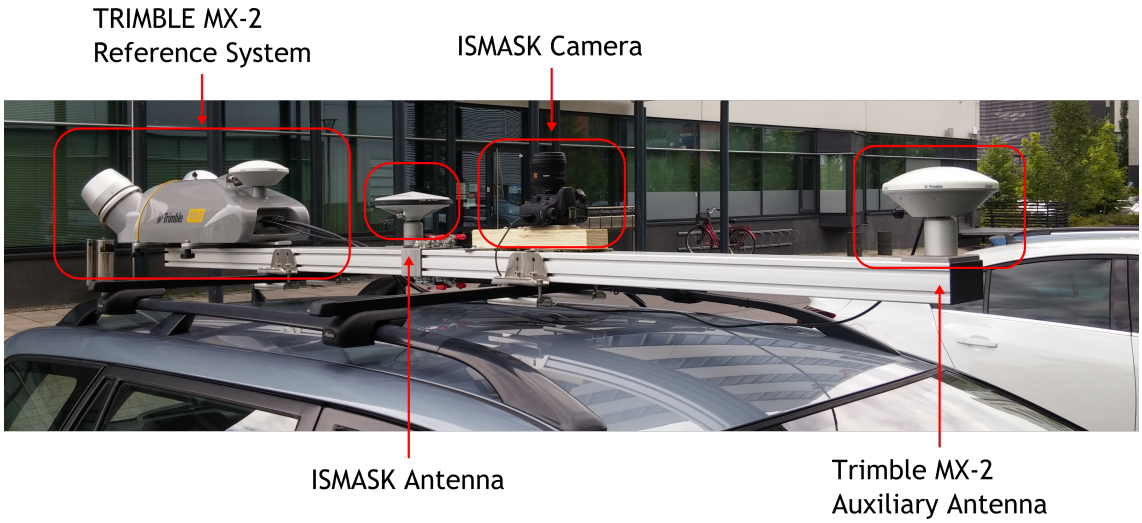

Figure 8: Test bench installed to the rover vehicle

The GNSS receiver used in the mobile measurement system is a Septentrio AsteRx-U HDC receiver. The receiver is a geodetic grade high accuracy GNSS receiver with 544 tracking channels and capability to track all known and future signals from GPS, GLONASS and Galileo. The receiver includes also advanced algorithms for multipath mitigation (APME+) and performance enhancement in elevated ionospheric activity. The receiver was powered in the test campaign by a leisure battery with capacity comparable to that of a car battery. The GNSS receiver was be set up to operate in rover mode while logging all observations and other available data with 1 Hz frequency to internal memory of the receiver in Septentrio Binary Format (.SBF file extension). SBF-files can be processed with Septentrio PPSDK as such and converted to RINEX format for RTKLIB processing.

In addition to logging the GNSS observation data the receiver was also set to use RTK positioning during the test campaign. RTK positioning is not actually needed for the analysis as it is done in post-processing; however, the RTK solution provides additional data on the GNSS navigation environment during the test campaign. The information was used to estimate the quality of the reference solution in order to be able to determine if short stops should be made in good visibility areas during the test runs to increase the quality of said reference solution. The outputs of the rover receiver are SBF files.

The measurement camera system consists of a Sigma Circular Fisheye lens and Canon EOS 70D DSLR camera body. The camera is attached to an angle iron which is in turn attached to a camera mounting plank. The camera needs to be mounted facing upward as accurately as possible. The camera is connected to the controlling laptop running an image capturing software which, in addition to image capturing, also logs accurate timestamps for the images. The images need to be taken as close to second ticks as possible in order to guarantee that generated signal masks match GNSS processing epochs. The method of time synchronization with image timestamps and other measurements was performed by using accurate time data from

the GNSS base receiver which was set up to act as a *network time protocol* (NTP) time server. Outputs of camera system are the sky facing images and timestamp file linking each image with accurate timing information.

The rough accuracy of image timestamping was evaluated by performing a small test. The test was done prior to the test campaign by receiving a NMEA $GPZDA sentence from the base station receiver via Internet in the same manner that was used in the test campaign. The received $GPZDA sentence contains UTC date and time information in centisecond (1/100 second) resolution. The received time information was displayed on a computer screen which was photographed with the camera in the same way as the image capturing was done in the measurement campaign. The difference in time information between the image timestamps and NMEA time was then compared. 20 image epochs were captured in total and the differences from these epochs are presented in figure 9.

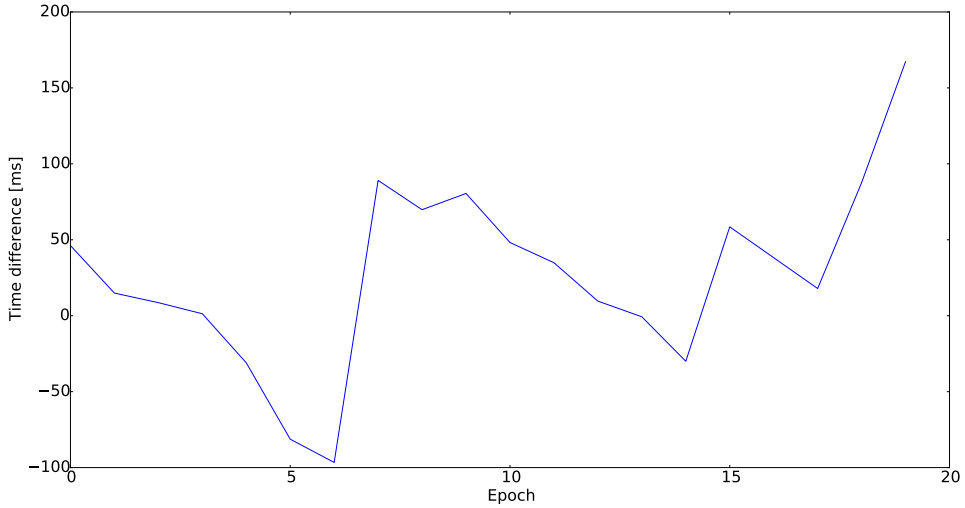

Figure 9: Timestamp accuracy with respect to GPS NMEA time

The mean difference between these time sources was 50 milliseconds while the extremes are below 200 milliseconds. A 200 millisecond timing error would cause an error of 4 meters when driving 80 km/h which would distort the generated mask by a potentially noticeable amount. The test velocities are mostly lower than 80 km/h and the errors are not as large as 200 milliseconds in general so the average location errors due to timing inaccuracy are likely well below meter level which is within limits of required accuracy for the mask location. The accuracy of this comparison method is not assessed in more detail as the goal of the test was to find out a rough level for the timestamp accuracies.

The Trimble MX-2 mobile mapping system equipped with a Trimble AP20 GNSS-Inertial navigation system acted as the reference system providing accurate positioning data from the test campaign. The GNSS-INS reference system was used not only to provide an accurate reference positioning solution but also to provide orientation information for the camera. The final outputs of the reference system

are the post-processed navigation solution shifted to the position of the ISMASK rover antenna including the vehicle position and orientation data.

The Trimble AP20 navigation system integrates a GPS and GLONASS capable GNSS receiver with an inertial navigation system in a tightly coupled manner. The reference system was equipped with two GNSS antennas which can be used to estimate attitude using a GPS azimuth measurement system (GAMS) and thereby provide a more accurate navigation solution. The system did not incorporate a distance measurement instrument (DMI)—often used in such setups—which could have assisted the navigation solution processing in heavily obstructed environments.

The collection of the reference data set some constraints to the test drives as the integrated GNSS inertial navigation system requires updates from GNSS data in order to estimate and correct the drift that occurs due to the accumulation of errors in the inertial measurements. The prevention of the drift phenomenon required to make stops when coming to areas with good sky visibility from the most heavily obstructed urban canyons. The drift related errors were most apparent on the dense urban areas where the longest sections with very poor satellite visibility lasted over five minutes.

The GNSS reference base station was set up on the roof of the SSF office building. The receiver served as a base station for the RTK processing. The GNSS receiver on the reference station—Septentrio PolaRx5—was set to observe and log all GNSS frequencies supported by the rover receiver: GPS (L1, L2 and L5), GLONASS (L1, L2 and L3), Galileo (E1, E5 and E6) at 1 Hz frequency. The used receiver is a geodetic grade receiver intended for base station usage. It features the same number of tracking channels and most of the same algorithms as the used GNSS rover receiver. Data was saved to the Septentrio Binary format as was the case also with the rover receiver.

The sky view from the SSF office building is very good with little to no obstructions. The coordinates for the base station mounting were calculated using the Trimnet CORS (Continuously Operating Reference Station) network operated by Geotrim guaranteeing reasonably accurate base station coordinates. The SSF office is located 10 kilometers from the Helsinki center and as such provides a typical RTK baseline length to densely built urban areas near Helsinki center. Rover RTK positioning during the test campaign required also differential corrections to be broadcast from the base station. The transmission of correction data was performed over the Internet. Output of the rover receiver is the SBF-file containing reference station data.

Prior to the actual measurement campaign preliminary tests were carried out with the ISMASK system in order to verify that all parts of the software and test bench are working properly. The tests included a similar system setup as was used in the test campaign, but with the difference that the test bench did not include the reference navigation system. The ISMASK system was found to be working properly and ready for the measurement campaign.

3.2 Test campaign

The test campaign was designed to be as comprehensive as possible with the project constraints in mind; the largest constraint is set by the available budget as the used reference system (Trimble MX-2) is very expensive to rent. In order to minimize the renting costs, the whole test campaign was conducted during a single day. The other limiting factor is set by the time available for the data analysis stage. A one-day test campaign provides reasonably extensive tests while still not exceeding project limitations.

The tests were conducted using pre-determined test routes in varying environments. The test routes cover urban areas, dense urban areas and the highway environment. Urban and dense urban areas especially contain a lot of difficult GNSS navigation environments mostly in form of urban canyons. The test campaign was divided to three separate test routes, each in distinct satellite navigation environment: highway, urban, and dense urban. The test campaign was also divided in time to two sessions—morning and afternoon—in both of which the three routes were driven once. Between the morning and afternoon drives there was an approximately three-hour gap for each of the routes. The test campaign contained therefore 6 test drives in all totaling roughly 2.5 hours of recorded data.

Although the urban canyons in Helsinki are not as steep as on larger metropolitan areas of the world, the visibility of the sky is very restricted on the dense urban route due to the majority of the route lying on narrow streets. Narrow streets with medium height buildings on the sides provide quite substantial sky-view obstructions resulting in a challenging satellite navigation environment. The location of Helsinki on high latitudes (approximately 60°N) further on reduces satellite visibility posing challenges for positioning performance.

The reference navigation system used for the ground truth reference solution generation imposes additional requirements for the test routes. The quality of the reference solution degrades in time due to accumulation of inertial measurement errors when the system is unable to provide GNSS solutions. In order to improve quality of the reference solution, places with good sky visibility were added to the routes in the most difficult navigation environments. Short stops in these good visibility locations allow the reference system to improve solution quality and mitigate the accumulated errors. These stops obviously make the route easier also for the ISMASK RTK system in addition to the reference navigation system. This has the side effect of improving 3D model based mask generation which requires initial position estimates. If the initial position is not an accurate one—as is the case with non-RTK solutions—the signal mask has a risk of being calculated on a wrong location. A wrong mask location will then in result degrade the accuracy of the filtered positioning solution potentially providing worse performance than an unaided positioning would and the effect propagates in time as degraded performance leads to subsequent masks being also calculated for erroneous locations.

An important distinction between the mask generated from images and from a 3D model is that the 3D model does not contain trees or other smaller than building sized obstructions. The foliage may serve as a very dominating obstruction for the

mask if there are large trees at the roadside and hence the image based signal mask will be very different to the one calculated from the 3D model. Different masks will cause the processed solutions to differ by amounts depending on the obstruction environment. The test routes were designed so that there are stretches where there are no trees or the trees are located in front of buildings and also stretches where foliage is quite extensive.

Two of the routes, namely the urban and dense urban routes, are located in central Helsinki while the third one—highway route—is located due west from the Helsinki center. Urban and dense urban routes contain mostly varying levels of urban obstructions while the highway route runs mostly on a highway providing a very different kind of environment from the urban areas. The highway is however an "urban highway" meaning that there are occasional bridges going over the street unlike the more rural highway stretches where such obstructions are very rare. The dense urban route is located for most parts in the Punavuori area while the urban route is located in Ruoholahti the neighborhood and surrounding areas. The highway route runs on Länsiväylä between Ruoholahti and Hanasaari.

The shortest baselines from the base station located at the SSF office building are approximately 4 kilometers and are located on the highway route. The longest baselines are on the dense urban route being just over 10 kilometers. These baseline lengths are sufficiently short to allow GNSS usage in obstructed areas but still long enough to serve as typical RTK use cases.

Punavuori is a densely populated neighborhood with narrow streets in the older part of the Helsinki. Buildings are mostly blocks of flats with less than 8 floors. The dense urban route, running in Punavuori, is the most difficult environment for GNSS positioning due to buildings obstructing a relatively high portion of the sky. Ruoholahti is a much more recently built area with higher buildings and wider streets providing an easier satellite navigation environment for the urban route than on the dense urban route. Länsiväylä is a highway running from Ruoholahti to Espoo. As a highway environment there are not many obstructions near the road and hence the GNSS positioning environment is the least demanding of the three routes.

The dense urban route runs on narrow streets in densely populated urban areas. The route is 8.7 kilometers in length, corresponding to approximately 30–40 minutes of driving. Speed limits on the area are mostly 40 km/h but the area has quite a lot of traffic lights and other traffic which will slow the average speed down. A map of the test route is presented in figure 10. The route is illustrated in the image with green color. The magenta arc marks 10 km distance from the base station at the SSF office building. The used background map is Bing Aerial map.

The route starts and ends at Hernesaari (southwest on the map) which is an area with good visibility of the sky and hence provides good PVT solutions to both ends of the route. This is done mainly in order to improve overall quality of the reference navigation solution as the system benefits from strong ties to good quality solutions. From Hernesaari, the route goes on to Punavuori and surrounding neighborhoods with a few stretches of parking lots and parks in between the urban canyon sections to improve performance of both reference solution and the RTK solution of the ISMASK system. An image of a parking lot in the middle of the route is presented in figure

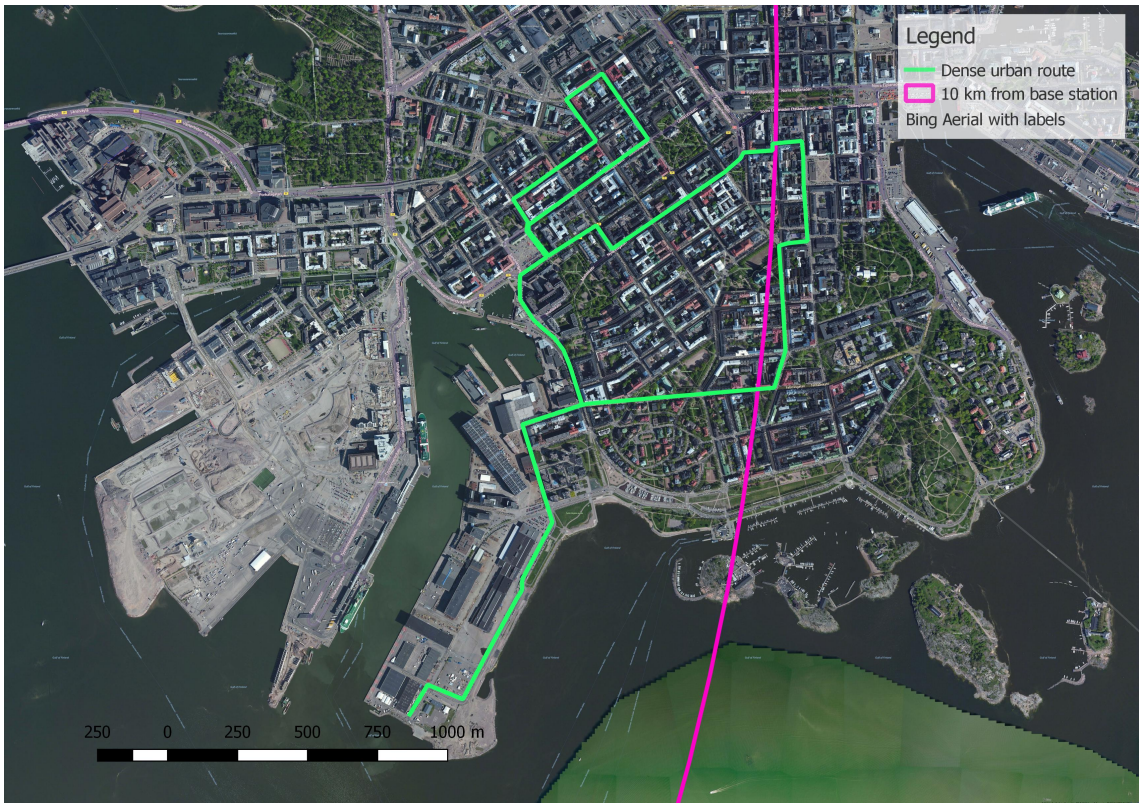

Figure 10: Test route in dense urban environment

11. The base image for the image is taken from Google Maps. A few minute stop in locations such as the one presented will serve to improve the quality of navigation solutions of both navigation systems in the middle of the route.

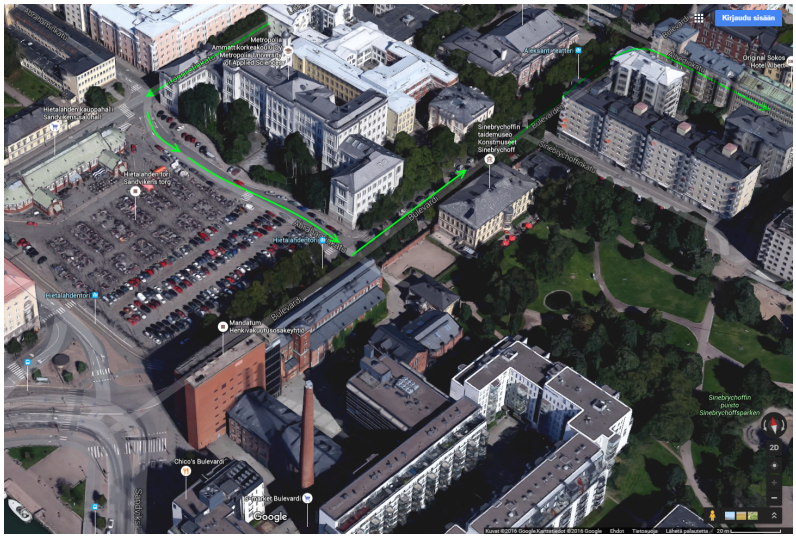

Figure 11: A parking lot with good sky visibility in the dense urban environment

Most of the buildings in the area are from 20 to 25 meters tall while the streets

are quite narrow. This combination limits visibility of the sky in street level to a narrow strip parallel to the street direction. An example mask constructed from the 3D model is presented in figure 12. As can be seen on the mask, there is a very limited visibility of the sky and the visibility is limited to only the direction parallel to the street resulting in non-optimal observation geometry no matter the locations of available satellites.

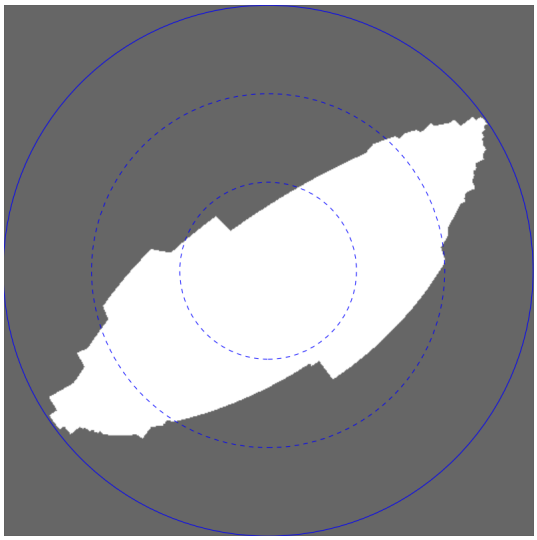

Figure 12: An example obstruction mask calculated from the 3D model on the dense urban route

Examples of urban canyons on the dense urban route are presented in figures 13 and 14 where green arrows depict driving direction. The base image in the images is taken from Google Maps. The urban canyons such as the ones in the example images—which cover most of the dense urban route—reduce GNSS navigation performance severely [32][25].

The urban route is intended to provide results in typical urban areas that are not extremely obstructed. The route acts as a middle ground between very easy environments on the highway route and the very difficult environment on the dense urban route. The urban route provides therefore a route with less demanding conditions for GNSS positioning than the route in Punavuori while still compromising a significant amount of obstructions. The route runs in the Ruoholahti neighborhood and extends to nearby neighborhoods. The urban route is in total 7.2 kilometers long with many similar characteristics and considerations as in the dense urban route. The route takes approximately 20 to 30 minutes to drive outside the rush hour. An overview of the route is given in figure 15 in which the route is illustrated with red color. The background map in the figure is Bing Aerial map.

Most of the buildings in Ruoholahti area are little higher than on Punavuori being approximately 20 to 30 meters. However, the wider streets in the area lead to less obstructed view of the sky in general. The environment in Ruoholahti provides therefore a little easier environment for GNSS positioning. An overview of the area is presented in figure 16. A sample obstruction mask can be seen in figure 17.

Figure 13: Example of urban canyons on the dense urban route

Figure 14: Another view of the urban canyons on the dense urban route

The route starts and ends at the parking lot by the sea on the north-western corner of the map providing a good visibility to at least half of the sky. At the beginning and the end of the route is a stretch with reasonably good sky view to provide good solutions to both ends of the route. There are similar areas with very good sky view at the middle of the route as is on the dense urban route—in fact the same parking lot is used at one point for this purpose.

The third test route—highway route—runs in an environment very different to urban and dense urban environments. There are far less obstructions close to the road and the obstructions are located farther away than on the other two routes. The driving speed is also considerably different to the driving speed in central Helsinki as the speed limit is 80 km/h compared to 40 km/h on the other routes. The route runs on Länsiväylä from Ruoholahti to Hanasaari and back. The entire round is

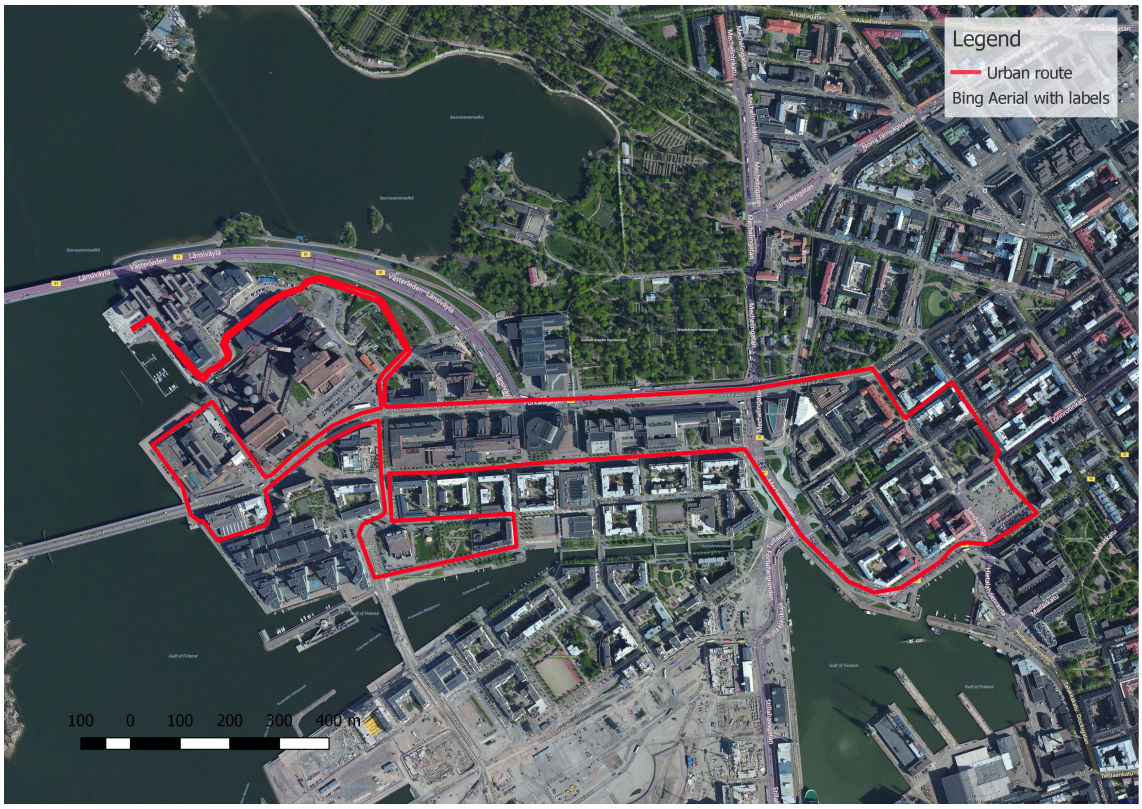

Figure 15: Test route in urban areas

approximately 8 kilometers in length taking approximately 10 minutes to drive. In order to get a sufficient amount of data the round was driven twice in both sessions.

The route is confined to only this 4 kilometer stretch of highway due to the publicly available 3D model covering only areas inside the city limits of Helsinki. Hanasaari is in fact already outside Helsinki in Espoo but the 3D model of Helsinki extends a little outside of the borders of Helsinki. An overview of the route is presented in figure 18 in which the route is illustrated with cyan on top of Bing Aerial map.

There is a very limited amount of obstructions near the highway, most of them being trees. There are, however, a few bridges and underpasses on the route that block the sky in its entirety. Due to these obstructions the route can be described as an urban highway, in contrast to rural highways which might be almost without any obstructions at all. An example mask calculated from the 3D model is presented in figure 19.

Unlike the other routes the highway route starts directly from the highway access ramp, not from a parking lot or other area with extraordinarily good sky visibility. This is due to the fact that the highway area is rather unobstructed and hence there is no need for a special section with a clear view of the sky at the beginning or end.

In addition to the test routes and navigation environments the test campaign planning required also selecting a suitable time to conduct the campaign. The selection of the test campaign date included taking into the account project time

Figure 16: An overview of the urban route

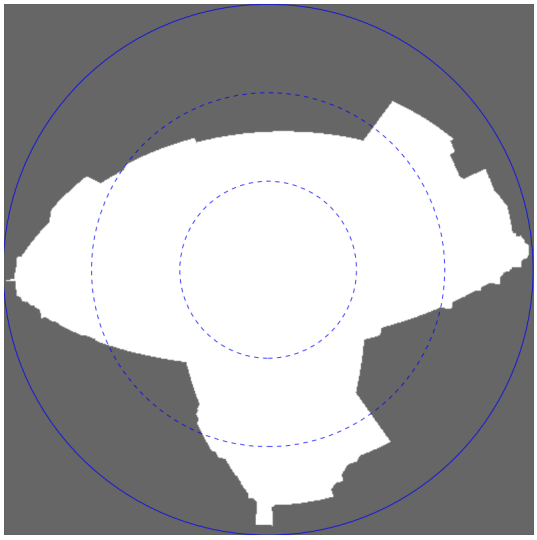

Figure 17: An example obstruction mask calculated from the 3D model on the urban route

frame, reference navigation system availability, and state of satellite constellations. The selection of suitable time of day on the test date depended on the availability of the reference system, traffic conditions on the test routes, and state of satellite constellations. The state of satellite constellations was inspected beforehand using an online GNSS planning service [39]. The status of satellite constellations was analyzed from the viewpoint of number and geometry of available satellites. The analysis of satellite geometry was concentrated on DOP and satellite elevations.

A suitable time for the test campaign was determined to be 6th of July between morning and afternoon rush hours. The number of satellites during that time was sufficiently good with also some satellites in high elevations for most of the time.

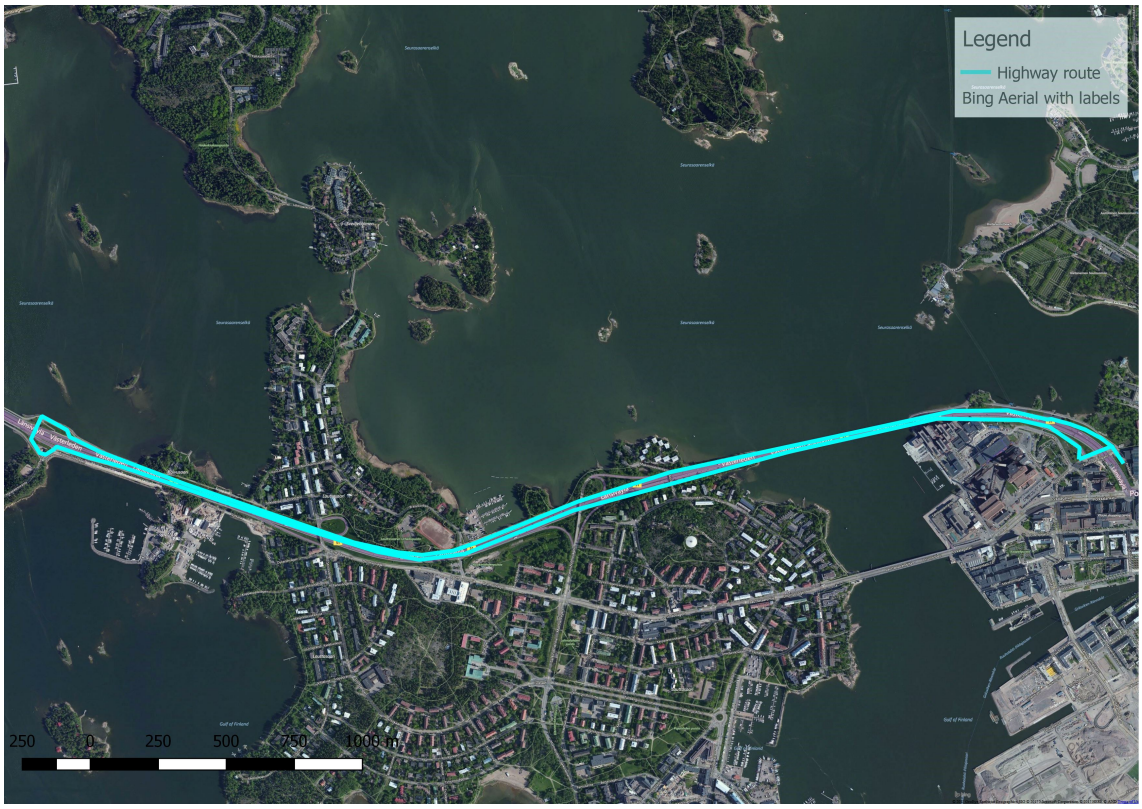

Figure 18: Test route in highway environment

There was also a drop in the number of available satellites in the middle of the day offering a variation in the satellite conditions.

The three routes presented here include steep urban canyons, shallower urban canyons, open environments, overhead bridges, various levels of vegetation, sections with single sided obstructions, and many other different kinds of satellite navigation environments. Altogether the routes provide a framework to properly assess the performance of the ISMASK system in different types of GNSS navigation environments. The considerations outlined here are taken into account in the data processing and result analysis.

The test campaign was performed on the 6$^{\text{th}}$ of July in 2016 as planned. The test campaign day began by picking up the reference navigation system and installing the whole test bench to the rover vehicle. The system was rented from Geotrim, a company providing surveying equipment and solutions. After system installation and driving to the test area, the reference navigation system had to be initialized by recording static and dynamical data with the GNSS and inertial sensors. The morning test session started with driving the highway route after which the urban route was driven before the session ending after the dense urban route.

After the morning session was completed there was a break before the afternoon session. The break gives time for the GNSS constellations to change between the two sessions providing information from the same routes with different satellite conditions. The afternoon session was performed in the same order as the morning session. After

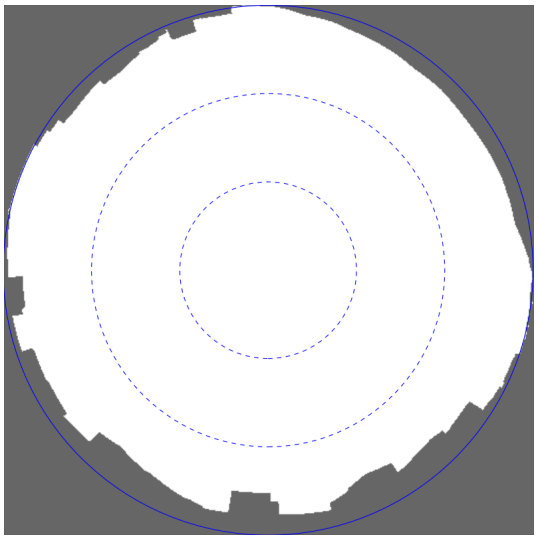

Figure 19: An example obstruction mask calculated from the 3D model on the highway route

the last afternoon route had been driven, the reference navigation system had to be initialized again before data recording could be turned off and the system be shut down. The day continued by a brief check on the recorded data in which no problems were detected.

The test campaign was deemed successful as all data collection went according to the plan and there were no unexpected problems. After the test campaign the recorded data was processed. The data processing is presented in detail in the next chapter.

3.3 Data processing

The test campaign and data processing are both prerequisites for the analysis of the ISMASK system. In the analysis phase, the solutions generated with ISMASK system in different configurations were analyzed. In order to quantify changes in positioning accuracy, the reference solution from the reference navigation system is used for ground truth comparison. The data processing constitutes therefore both processing of ISMASK data and the reference solution.

Before the ISMASK data could be input to the software, it had to be pre-processed. The pre-processing step includes also an inspection of the collected data in order to detect possible problems in the measurement campaign. Verification and analysis of the quality of the collected data is also a requirement to guarantee reliability of the obtained results. The actions done in the pre-processing phase differed depending on the recorded data in question.

The verification checks done in the pre-processing phase should be regarded more like a rough control check to detect possible errors, be they systematic errors or simple mistakes. Most of the checks are in no way valid methods for validating scientific tests but they are all designed to reveal some possible error sources. These checks

should, however, not be seen as complete validation of the ISMASK system and the test data. Some of the checks are also intended to get a rough estimate of subsystem accuracies and these tests and their results should be viewed as approximations of the real accuracy.

The following data was collected in the measurement campaign:

- base station GNSS data (Septentrio Binary Format, SBF), one file for the whole day
- rover GNSS data (SBF), one file for each of the six test drives
- reference navigation solution obtained from GNSS-INS reference system (Binary Applanix format), one file for the entire test duration
- sky view images taken with the camera system (JPG), one image for each second of the test drives
- image timestamp files (ASCII text file), one file for each of the six test drives.

In addition to actual test data the following logs and documents were also recorded:

- test record documents (Word document) for each six test runs
- measurements of the lever arms between rover GNSS antenna and the reference point of the GNSS-INS reference system, common for the whole test day
- a time synchronization log for the operating laptop, one log file for whole test day.

The ISMASK test software requires also the following input data which could have been obtained before or after the test campaign:

- digital surface models of the test areas
- GNSS antenna calibration files (ANTEX).

The test data was divided to six separate datasets according to the driven test routes. The datasets had individual rover GNSS files, images, timestamp files, and test record documents while the base station GNSS data, reference solution, lever arm measurements, time synchronization logs, digital surface models, and antenna calibration files were common for all datasets.

There are two distinct types of test data: data input to the ISMASK software and data relating to the reference solution. The ISMASK software outputs a PVT solution which can then be compared to the reference solution in the analysis phase. All other collected data besides the reference solution is input to the developed software. The measured lever arm information is, however, used in the reference solution processing in order to transform the reference solution to the same location as solutions computed by the ISMASK software. The inputs of the ISMASK software are presented in figure 20.

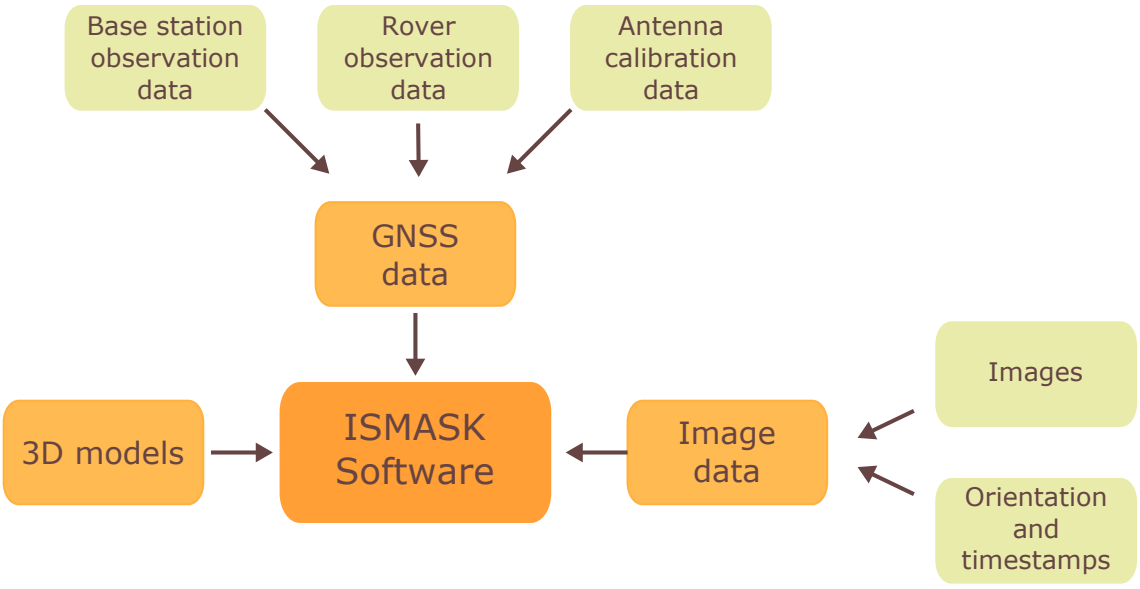

Figure 20: Input data for the ISMASK software

The pre-processing started with reviewing all recorded logs and other records and storing these files for documentation purposes. No inconsistencies were found from the logs or the test record documents.

GNSS data was recorded in SBF-format which is native format for the receivers and also compatible with the Septentrio PPSDK. RTKLIB requires, however, the SBF files to be converted into the RINEX format. The obtained GNSS files were also post-processed with their respective software to obtain sample solutions. The solutions were inspected to verify that the collected GNSS data is not faulty.

From the recorded GNSS data it was observed that there was indeed a drop in the number of observed satellites between the morning and afternoon sessions as was to be expected from the test campaign plans. The change in the satellite environment was observed to make the positioning environment in the afternoon tests more difficult than in the morning. The number of satellites tracked by the base station receiver is presented in figure 21. The figure displays the number of satellites for four different sets of constellations sampled at 30 second interval to reduce noise in the figure. The different sets are GPS only, GPS and GLONASS, GPS and Galileo, and all three constellations together. The different sets are presented to demonstrate satellites used by different software. GPS and GLONASS are used for reference navigation solution processing and for Septentrio PPSDK RTK processing while the non-RTK solutions produced by Septentrio PPSDK utilize GPS, GLONASS, and Galileo. RTKLIB uses GPS, GLONASS, and Galileo all, but due to GLONASS inter-frequency biases is unable to fix GLONASS integer ambiguities. The figure presents also bars indicating duration of each test run.

All of the satellites visible on the base station receiver are not obviously available at the rover receiver but the general status of the constellation is fairly similar. The lowest number of available satellites was during the afternoon sessions on highway and urban routes. At the time of the afternoon session on the dense urban route the

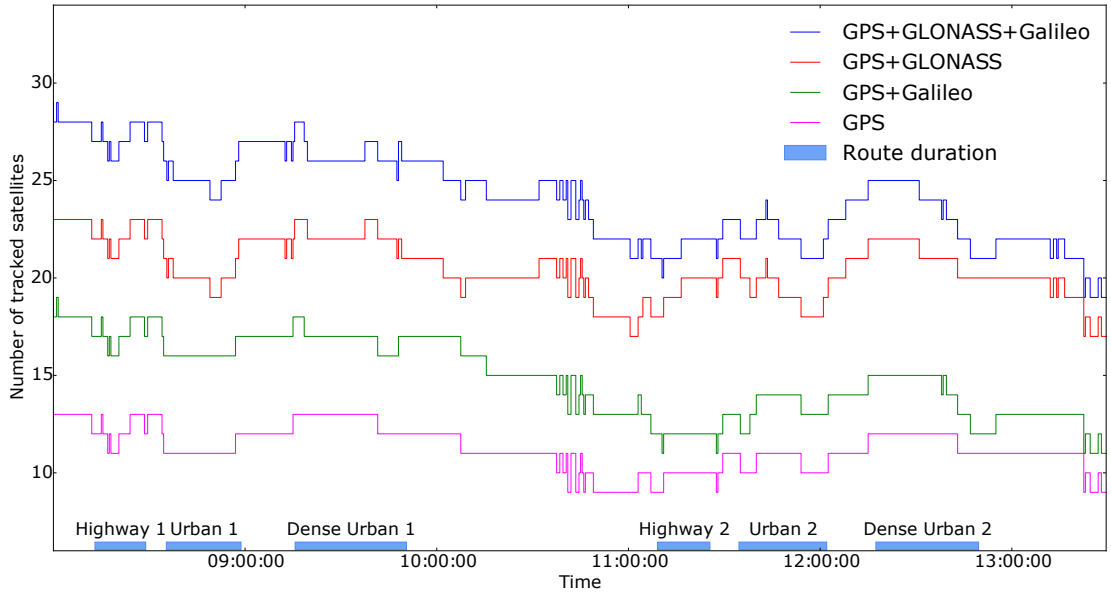

Figure 21: Number of satellites tracked on base station receiver during the test campaign

number of satellites had improved from the lowest values which occurred between 11:00 and 12:00.

The sky-view images were only visually checked to verify that there have not been any major problems with the camera system. Image lighting and colors were also visually checked to be on suitable level for the sky detection. The sky detection algorithm was tested by picking a random set of images from the tests and drawing obstruction masks on top of the images which were visually analyzed. From the analysis, it was concluded that the sky detection works in the majority of cases, leaving some bright façades and other difficult-to-detect targets unmasked.

Image timestamp files were analyzed by verifying that the image timestamps do not deviate too much from the second ticks in which the GNSS observations are recorded. All of the timestamps were found to be closer than 0.1 seconds to the second ticks while the majority of the timestamps were less than 0.015 seconds from the ticks.

The timestamp accuracy was assessed by comparing masks generated from images and 3D model. The 3D model based mask is located at the RTK solution which is accurate to centimeter level when a good quality RTK solution is obtained. If the image based obstruction mask resembles closely the 3D model based masks in an area where there are no clear distinctions between the masks such as foliage or buildings missing from the DSM, the timestamp accuracy could be deemed of sufficiently quality. Suitable areas for this kind of comparison were selected from the test data and the masks were compared. The compared masks were determined to resemble each other well enough to conclude that the accuracy of image timestamps is on a suitable level for the mask generation.

The PVT solution output by the ISMASK system can be used as a secondary

attitude source if the limitations of the single antenna attitude determination are taken into account. An approximation of the heading and pitch angles for the rover vehicle can be determined from the GNSS velocity vector but the quality of such an attitude computation is very low compared to the INS attitude. The single antenna attitude solution also lacks ability to approximate the roll angle. The GNSS derived attitude was tested prior to the actual test campaign in order to be able to test mask generation beforehand. The yaw angle can be estimated from the horizontal direction of the velocity vector. The pitch angle can be estimated from the angle between velocity vector and horizontal plane. Determination of the roll angle would require a two antenna setup which is not available in the ISMASK system.

The velocity vector derived orientation was used to provide a very rough control for the MX-2 attitude solutions, mainly to check that the definition of the orientation angles matches the system implemented in the ISMASK software which was tested in preliminary tests. The obtained headings, i.e. yaw angle, from both reference solution and GNSS velocity vector were mostly consistent in the datasets if the inferior precision of the GNSS derived attitude is taken into account, except in the morning urban test. The problems with the morning urban data were related to a large number of the angles being wrongly interpolated. The attitudes had the best matching in the highway datasets where the satellite visibility is extremely good compared to other routes. The pitch angles were consistent between the two attitude solutions probably due to the small magnitude of changes in pitch angles on the relatively flat test environments.

From the comparison of the two attitude sources it was concluded that the coordinate transformations implemented in the ISMASK software are compatible with the attitude provided by the reference navigation system. The comparison of 3D model and image based masks done to control image timestamp accuracy provides also another control for the attitude as the image based mask requires the attitude solution. In all tests no notable problems were observed with the reference system attitude solutions and the solutions were determined to be usable for all test datasets.

The processing of the reference solution required the data collected with the reference navigation system and lever arm measurements as inputs to the processing. The reference solution processing was performed in post-processing mode after the test campaign.

In order to compare solutions made at different physical locations—at location of the ISMASK antenna and at the location of the reference point of the reference navigation system—the lever arms between the locations needed to be measured. The measurements had also to be verified in order to guarantee reliability of the transformation parameters between the solutions. A schematic of the positioning sensor installation and lever arm components (distances in mm) is presented in figure 22. The gray bar in the bottom is the MX-2 mounting bar to which all sensors are attached. The big box on the right side is the MX-2 reference system on top of which is its primary antenna. The left-most antenna in the schematic is the MX-2 auxiliary antenna. The antenna in middle is the ISMASK GNSS antenna.

In addition to the sensors illustrated in figure 22, the camera system was also set up to the mounting bar in the space between the two leftmost antennas. The

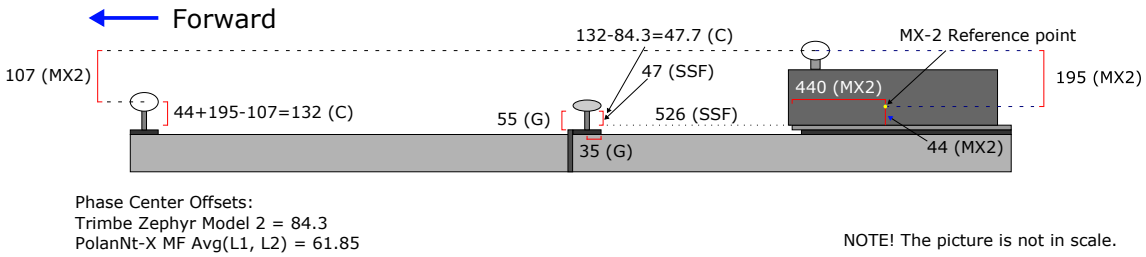

Figure 22: ISMASK Test bench sensors and lever arms

camera location is however not of interest in lever arm sense as the image mask cannot be translated to a different origin. Precise comparison of the 3D model and camera based masks would indeed require translation of the 3D model mask to the camera location but such a comparison was not deemed necessary as the masks were successfully compared with required reliability using less precise methods.

Reference solutions are expressed with reference to the MX-2 reference point which is depicted in yellow inside the MX-2 casing in the schematic. All measurements made by the MX-2 system are translated to this point from which the resulting solution can be translated to a different origin with the help of lever arm measurements. For the ISMASK test system all GNSS measurements refer to the antenna reference point of Septentrio PolaNt-X MF antenna which is located at the bottom of the antenna mount. The lever arm from the MX-2 reference point to bottom of the ISMASK antenna mount is then needed to transform reference system solutions to the same point as ISMASK PVT solutions.

All sensors are mounted to rails embedded in the mounting bar which results in all sensors being on a same line in the direction of the bar and no across-bar component in the lever arm. The vector from the MX-2 reference point to the top-left point of the lighter gray sheet under the casing is also known from the information provided by the system manufacturer. So in order to measure the distance, the measurements have to be made from the top-left point of the lighter gray sheet to the bottom of the ISMASK antenna mount and only in directions corresponding to horizontal and vertical directions in the schematic.

The schematic presents measurements from four distinct sources: MX-2 system documents, measurements by SSF employees during the test campaign, measurements by employees of the company renting the MX-2 system, and values derived from other measurements. The source of given measurement is marked in parentheses after the measurement; MX-2 referring to dimension documents, SSF to SSF employees, G to employees of Geotrim, and C for the distances computed from other measurements. Some of the measurements are redundant and used only for verification purposes.

The lever arm from MX-2 reference point to the ISMASK reference point can be calculated from the illustrated measurements using the following coordinate frame convention: X, Y, Z coordinate frame, origin of which is in MX-2 reference point, X-axis pointing in vehicle forward direction, Y-axis vehicle right, and Z-axis down. The lever arm X component is then given by $\Delta X = 440\,\text{mm} + 526\,\text{mm} + 35\,\text{mm} = 1001\,\text{mm}$. The Y component $\Delta Y = 0\,\text{mm}$ due to the used mounting system in which all sensors

are aligned. Finally, the Z component is given by $\Delta Z = 44\,\text{mm} - 47\,\text{mm} = -3\,\text{mm}$. The Z component can be calculated also in another way by utilizing known dimensions of the mountings used for both the MX-2 auxiliary antenna and the ISMASK antenna. The alternative method yields a Z component of $\Delta Z_2 = 44\,\text{mm} - 47.7\,\text{mm} = -3.7\,\text{mm}$. The difference of less than a millimeter is sufficiently small that the lever arm measurements can be seen to be in agreement.

The lever arm was used to translate the reference solution to the same location as the ISMASK position solution. The solutions of the two positioning systems—ISMASK GNSS and GNSS-INS reference system—can therefore be compared in order to guarantee that the systems output correct coordinates and that the lever arm is properly computed. A small test with good quality solutions was conducted to compare the two solutions. The tests were carried out with 10 epochs of data from each test dataset. The epochs were selected on the basis of being fixed RTK solutions with small standard deviations in at least reasonably open areas where the vehicle is stationary at the beginning of each test run. The resulting distances between the two solutions were observed to be within the limits of standard centimeter-level RTK positioning accuracy. It was concluded that the lever arm is correctly computed and the reference solution was properly transformed to the desired location.

The reference navigation solution from the Trimble MX-2 system had to be processed with Applanix POSPac software using Virtual Reference Station data obtained from the Trimnet VRS service operated by Geotrim. The measured lever arm can be configured in the POSPac software which then uses the information to translate the solution from the MX-2 reference point to the bottom of the ISMASK antenna mount. The output of the POSPac software is an ASCII solution file in which each epoch has timestamps, position coordinates, orientation (Tait-Bryan) angles, ENU (east, north, up) velocity components, ENU position standard deviations, and orientation angle standard deviations.

The most important use for the reference solution is the ability to evaluate the accuracy of obtained position solutions and to distinguish correct and incorrect RTK fixes from each other. Correct fix refers here to a fixed RTK solution in which integer ambiguities are fixed to correct integer values contrary to incorrect fix in which the integer ambiguity estimates are erroneous. The detection of fix correctness allows assessment of whether or not a possible increase in the amount of fixed solutions is indeed an improvement in the positioning performance or just an increase in the number of false fixes. In order to detect if a fixed solution is indeed a correct one a limit for the estimate of reference solution precision is in this study required to be at least on the 10 cm level. The limit of 10 cm is derived from the carrier wavelengths which reside in levels of roughly 20 cm.

The quality of the obtained reference solution can be assessed from the standard deviations of the reference solution. The integrated GNSS-INS navigation system should be able to provide reasonably accurate estimates of the solution quality, especially on the level required to detect fix correctness. The position standard deviations of the reference positioning solutions are presented in figure 23. The 10 cm limit for solution standard deviation is shown as a red line in the figure. From the figure, it can be seen that almost all of the highway and urban routes are below the

limit but on the dense urban route the quality of the reference solution is sufficiently good only on roughly half of all the epochs.

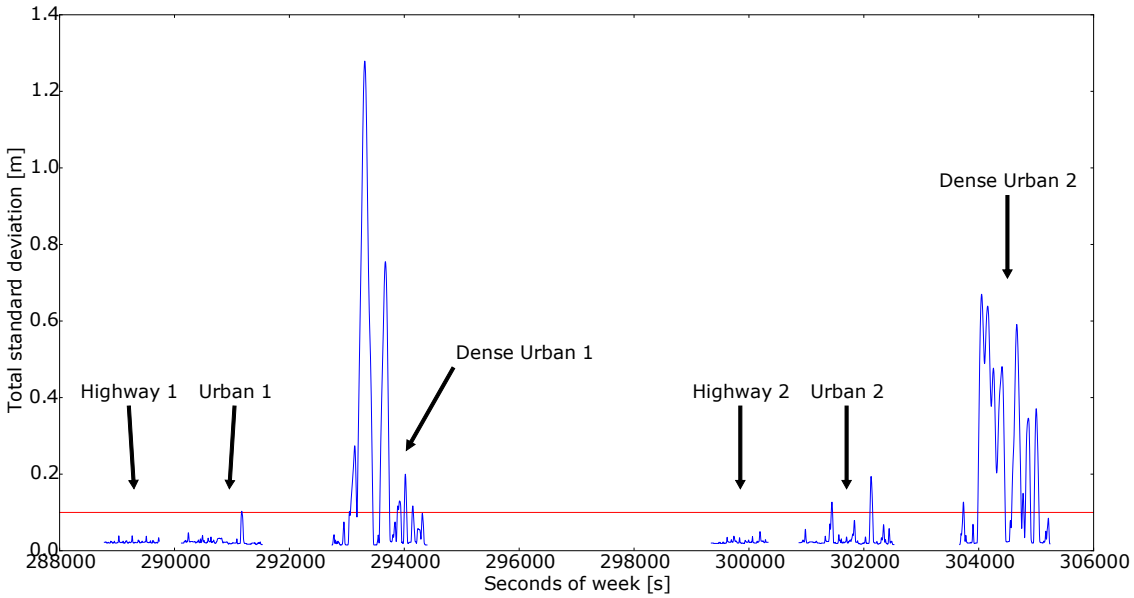

Figure 23: Total position standard deviation of the GNSS-INS reference solution

While the standard deviation of the obtained solutions was approximately a few centimeters for large part of the tests, the satellite visibility conditions on the dense urban route were too demanding for the reference system to provide accurate solutions for all epochs. In the analysis phase the reference solution is used only for the epochs where the standard deviation is below the defined limit of 10 cm.

After the pre-processing of ISMASK data was complete, the data was input to the ISMASK system for the actual processing. The ISMASK processing—from here on just processing—is divided to two distinct parts, first of which uses Septentrio PPSDK for the GNSS processing and the second using RTKLIB for the GNSS processing.

The ISMASK software allows multiple different methods for the obstruction mask generation. The mask can be generated either from a digital surface model, a sky-view image, or from both of these. There are also two distinct types of digital surface model derived masks: with and without the use of likely multipath interference masking. The combination masking is done however only without the multipath interference masking.

These mask generation modes result in five different test modes for the ISMASK processing software:

- no masking
- digital surface model derived masking
- digital surface model derived masking without multipath interference
- sky-view image derived masking

- combined masking of sky-view image and digital surface model masks.

The different mask generation methods are analyzed with the goal being identification of an optimal mask generation method and to assess feasibility of the different methods. The five possible test modes with two different GNSS processing software packages totals 10 different solutions for each dataset. All 10 solutions for six different test drives are incorporated into the analysis phase totaling 60 different solutions. These 60 solutions provide a great number of different scenarios and processing methods from which the performance of the Intelligent Signal Masking for GNSS RTK system can be evaluated.

3.4 Analysis

The data collected in the test campaign with the ISMASK test bench and processed using the ISMASK software was analyzed with the goal of determining the feasibility of the developed masking system and effects of different masking methods. The analysis incorporates in addition to the solutions output from the ISMASK system, also the reference solution from the reference navigation system.

The analysis of different masking methods required individual processing for the data using all the masking methods. The processing had to be carried out in exactly the same manner, except for the masking method, for all the methods in order to obtain reliable results for comparison of the different methods. The methods were analyzed separately in the context of the two used GNSS processing software packages. From the solutions provided by different masking configurations the effects of different methods can be assessed.

Overall feasibility of the ISMASK system was evaluated also separately for the two GNSS processing software packages. The overall feasibility assessment required information of the different masking methods, especially the information of the most effective masking method. The comparison between the most effective masking method and the unmasked solution provides an indication of how feasible a real commercial implementation of the ISMASK system would be.

The performance of the navigation system was analyzed in this study in the context of positioning accuracy and availability of positioning solution. The availability was analyzed in particular in the context of fixed RTK solution availability. The availability analysis was concentrated on the RTK solution due to the ISMASK project being a study of RTK navigation and assessment of fixed solutions was done due to one of the potential improvement areas for ISMASK system being ambiguity resolution. The potential for fix percentage improvement is due to filtering of NLOS satellites which may hamper the integer ambiguity resolution.

Improvement of the positioning performance in terms of accuracy is analyzed in almost all similar studies focusing on the GNSS performance improvement [45][44][23][27][46][63][58][29]. Fix percentage has also been used in the assessment of the fisheye camera based satellite filtering system by Suzuki and Kubo [58].

Accuracy is characterized in this study by *root mean square* (RMS) 3D position error calculated from the distance between the obtained ISMASK solution and the

GNSS-INS reference solution. RMS is an established metric for positioning accuracy used in many similar studies to characterize positioning accuracy [45][44][27][63]. The positioning error is determined to be the distance between the two solutions for a given time from which the 3D RMS error is calculated as

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\Delta E^2 + \Delta N^2 + \Delta U^2)} \tag{25}$$

where n is the number of epochs and ΔE, ΔN, and ΔU are east, north, and up distances between the ISMASK and reference solutions, respectively. The 3D RMS error is calculated for all epochs for which the quality of the reference solution is at an acceptable level.

Fix availability is characterized by the percentage of fixed RTK solutions out of all epochs. A greater fix percentage can generally be seen as an indication of better positioning performance. There is, however, a problem with fix percentage as a performance metric as it doesn't disqualify solutions in which the GNSS processing software resolves integer ambiguities wrongly. A wrong integer ambiguity resolution causes a ranging error corresponding to at least the size of the carrier wavelength, approximately 20 cm for GPS L1.

In order to provide information about correctly fixed solutions, a correct fix percentage was determined using solution fix information in combination with positioning error information derived from the reference solution. Correct fix percentage is the percentage of all epochs that are fixed RTK solutions and determined to be correct in a sense that the integer ambiguities are fixed to correct values. The correctness is determined from the distance to the reference solution and assumed to be correct if the distance (or position error) is less than 10 cm. The correctness is determined only for the epochs for which the reference solution is available with sufficient precision.

Fix percentage F is calculated as

$$F = \frac{N_{\text{fixed}}}{N_{\text{total epochs}}} 100\% \tag{26}$$

where N_{fixed} is the number of epochs with fixed RTK solution and $N_{\text{total epochs}}$ is number of total epochs.

Correct fix percentage F_{correct} is similarly given by

$$F_{\text{correct}} = \frac{N_{\text{correctly fixed}}}{N_{\text{total epochs}}} 100\% \tag{27}$$

where $N_{\text{correctly fixed}}$ is the number of epochs which are fixed RTK solutions with 3D positioning errors less than 10 cm.

Calculating the fix and correct fix percentages as presented above has a benefit of having the same number of total epochs and therefore comparable results. The downside is that the calculation assumes that the solutions for which there is no sufficiently accurate reference solution are regarded as not correct solutions although the correctness is in fact undetermined. An alternative solution would be to use

number of total epochs with available reference solution instead of $N_{\text{total epochs}}$ but this would lead to different number of total epochs and possibly higher correct fix percentage than the fix percentage. Incorporating only the solutions with available reference solution to also the fix percentage would in the other hand discard valid data from the analysis while adding an additional third fix percentage related measure would unnecessarily complicate the analysis.

The assumption that fixes without available reference data are not fixed or are fixed incorrectly is not totally irrational, as the accuracy of the integrated GNSS-INS navigation system is of higher level than the standard RTK accuracy. In the case where the reference system is unable to provide a solution with fixed RTK accuracy level it is unlikely that GNSS-only solution would reliably achieve this accuracy level. However, some of the epochs without reference data available would likely have been determined as correct if the reference solution would be of sufficient quality on the entire route. Moreover, the unavailability of the reference data is not a real concern in highway and urban datasets, but in the dense urban datasets approximately half of the epochs do not have a reference solution with sufficient quality. In total in all datasets approximately 20 % of all epochs do not have an acceptable reference solution available.

RMS and fix percentage, especially correct fix percentage, are all related to accuracy of the positioning system and therefore characterize similar aspects of the positioning performance. In the sense of accuracy, RMS is sensitive to a small number of large errors due to squaring of individual errors. Fix percentages in turn are more related to long term steady level accuracy as fixed solutions are on average more accurate than float or non-RTK solutions.

The analysis will therefore contain two distinct types of metrics: the ones that require a reference solution and the ones that do not. The problem of reference solution cover dilutes the reliability of reference related metrics on the dense urban route where a large portion of the total data is without viable reference solution. Despite the problems there is however a large amount of data even from more difficult environments with available reference solution but not up to the level that was planned. Fix percentage is not suffering from these problems as it is not dependent on the reference solution.

The data collected in the test campaign and processed afterwards were analyzed according to the methods described in this section. The results from the analysis are presented in the next chapter.

4 Results

The effects of the ISMASK system presented in chapter 3.1 on the RTK positioning performance was analyzed with the methods described in chapter 3.4. The results of the analysis are presented in this chapter. The results are presented first as overall results from all environments after which following sections present the results in terms of the individual navigation environments categorized according to the test routes.

4.1 All environments

The entire test data from all test routes consists of approximately 8200 epochs recorded with 1 second intervals. The data provides a wide variety of different navigation environments from the three test routes. The combined results from all test datasets for RTKLIB solutions are presented in table 1 and Septentrio PPSDK solutions in table 2. The tables contain 3D RMS errors, fix percentages, and correct fix percentages as presented in chapter 3.4 from all test datasets for all five test modes. Out of epochs approximately 1600 are missing a reference solution of the required quality. The analyzed solutions are compared in the tables using three different statistical quantities presented in section 3.4.

Table 1: RTKLIB results of all datasets

	No Mask	Model	Model No Multipath	Image	Combined
3D RMS Error [m]	6.05	4.72 (-22%)	6.2 (+2%)	4.33 (-28%)	3.95 (-35%)
Fix %	37 %	45 % (+22%)	45 % (+22%)	47 % (+27%)	49 % (+32%)
Correct Fix %	36 %	43 % (+19%)	42 % (+17%)	45 % (+25%)	46 % (+28%)

Table 2: Septentrio PPSDK results of all datasets

	No Mask	Model	Model No Multipath	Image	Combined
3D RMS Error [m]	1.43	2.00 (+40%)	1.42 (-1%)	1.45 (+1%)	1.30 (-9%)
Fix %	70 %	65 % (-7%)	68 % (-3%)	67 % (-4%)	67 % (-4%)
Correct Fix %	67 %	63 % (-6%)	67 % (0%)	65 % (-3%)	65 % (-3%)

The tables 1 and 2 include 3D RMS errors, fix percentage, and correct fix percentage for all used test modes in which the masking software was run: no masking, DSM based masking, DSM based masking without multipath interference masking, sky-view image based masking, and combined DSM and image masking. For each measure except the ones for no masking mode there is also a percentage in parentheses indicating the change with respect to the same measure of unmasked solution. All data is presented in similar manner for both used GNSS processing software RTKLIB and Septentrio PPSDK.

From the results it can be seen that there was a clear difference between solutions processed using RTKLIB and Septentrio PPSDK. Septentrio PPSDK was able to

obtain a substantially higher number of fixed solutions and the accuracy of the obtained solutions was generally significantly better. An important distinction between the processing software packages is that due to the different processing methods and algorithms along with different sets of used satellites, there were large differences on the initial unmasked positioning performance and on the effects of signal masking. Also, masking does not appear to improve solutions when using the PPSDK software.

The quality of obtained solutions with Septentrio PPSDK was in general significantly better than the quality of solutions obtained with RTKLIB; one large factor to this is the inability of RTKLIB to handle GLONASS inter-frequency biases resulting in a much smaller set of satellites for which integer ambiguities can be fixed, observed in figure 21. Another large contributor to the worse quality of the RTKLIB solutions is that the proprietary algorithms developed by Septentrio are likely optimized to Septentrio hardware and include possibly advanced proprietary processing algorithms that are absent in RTKLIB. Septentrio PPSDK likely does some kind of a signal filtering using information such as multipath estimates, signal strength, and satellite elevations obtained by receivers using algorithms that are not used in RTKLIB.

The obtained results are first discussed in terms of each independent processing software and after that in terms of different masking modes. The goal being assessing how the ISMASK system does affect RTK positioning performance for both processing software packages and how do the different masking modes contribute to these effects.

The effects of signal masking and filtering on the positioning performance vary between the two processing software packages: signal masking reduces fix availability for Septentrio PPSDK while the masking improves fix availability for RTKLIB. Positioning accuracy in terms of RMS position errors was improved for both software packages but the improvement was much greater for the RTKLIB solutions—with best results contributing 35 % decrease in RMS error for RTKLIB and 9 % decrease for PPSDK.

Septentrio processing software is most likely able to filter out most of the signals causing multipath interference and possibly also NLOS reception errors by using built-in signal selection and filtering algorithms in RTK processing. Septentrio's selection and filtering step leaves fewer satellites to be filtered out by the ISMASK software. The additional filtering layer of the ISMASK software might also disrupt Septentrio's own algorithms and thereby contribute to negative effect in the number of fixes.

The APME+ technology included in the used receivers apply multipath interference compensations to observations at receiver level meaning that these corrections are included also in RTKLIB processing. However, the multipath estimates themselves are not present in RTKLIB processing but the information could be used by Septentrio PPSDK to perform additional filtering. If Septentrio's selection and filtering algorithms have already excluded or corrected a portion of the erroneous satellites or signals, the benefits of ISMASK filtering might be diminished and even contribute to lower performance.

A histogram of the RTKLIB positioning errors for all six datasets processed with and without signal masking is presented in figure 24. The used signal masking in the

figure is the combined image and 3D model method. The position errors are divided into 6 groups: below 5 cm errors, from 5 to 10 cm errors, from 10 cm to 1 m errors, from 1 m to 5 m errors, errors larger than 5 m, and epochs without a PVT solution. As we can see from the figure, the ISMASK masking system is able to reduce the number of the largest positioning errors by more than 50 % and increase the number of best quality solutions while the number of medium quality solutions remains on relatively identical levels. The improvement in the distribution of errors is in line with the increase in positioning accuracy observed in RMS positioning errors.

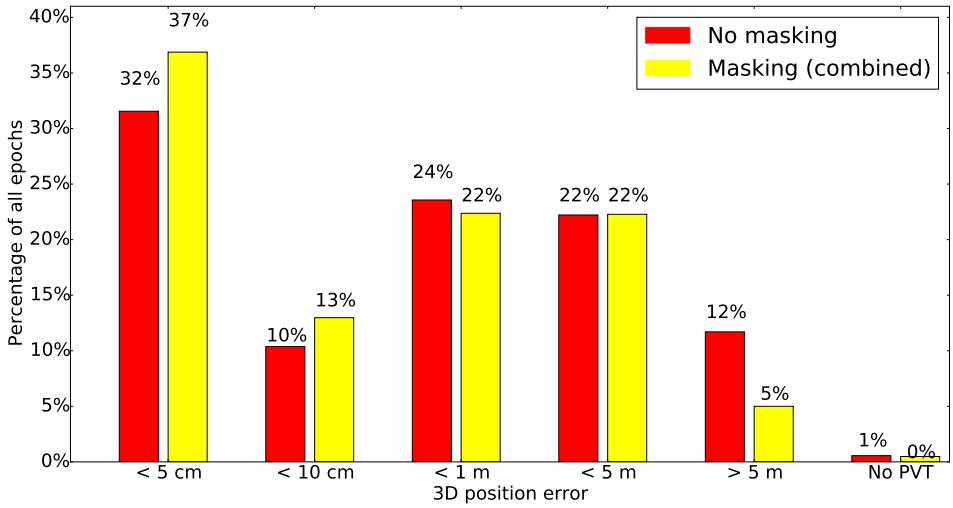

Figure 24: Histogram of RTKLIB 3D position errors

A similar histogram for the Septentrio PPSDK processed solutions is presented in figure 25. While there is an indication of improved positioning accuracy for Septentrio positioning by the improvement in RMS error, the histogram shows no real benefits for satellite masking. It can be seen on the histogram that the masking decreases the number of smallest errors and increases numbers of larger errors but the differences are very small and the distribution of the errors does not certainly improve with the masking.

In the light of presented data, the ISMASK satellite masking and filtering system is less efficient in improving the performance of RTK positioning when using software utilizing similar processing algorithms to the Septentrio PPSDK than for RTKLIB type software. The system has improved the accuracy of positioning for both software packages, but the effect on availability of fixed solutions is positive only for RTKLIB solutions and negative for Septentrio solutions. The total effect in RTK performance for Septentrio software is substantially closer to neutral than the positive impact observed on RTKLIB.

The results obtained with different test modes can be seen from the tables 1 and 2. For RTKLIB all used metrics support using the combined signal masking. The combined masking takes advantage of both image based and 3D model based masking data utilizing the data in a way that is designed to be less strict than either of the

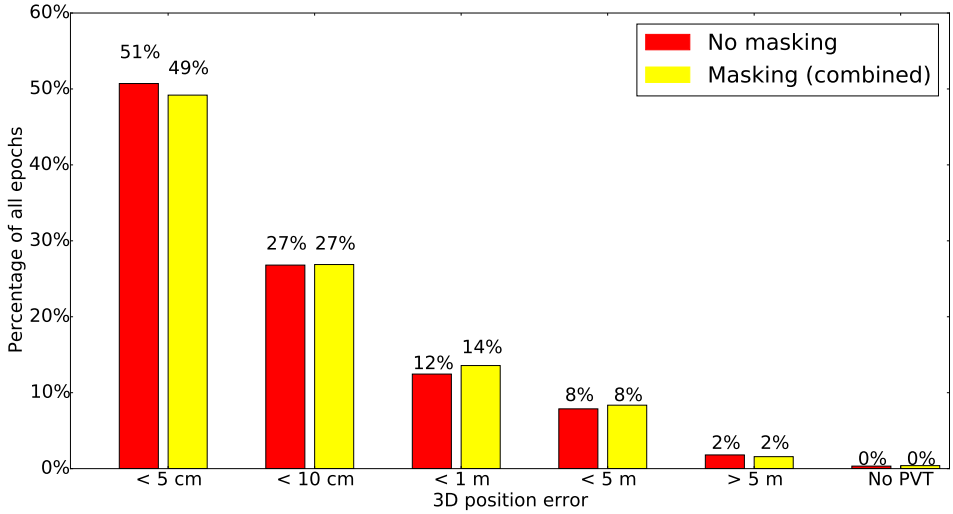

Figure 25: Histogram of Septentrio PPSDK 3D position errors

masking modes alone. Reducing the number of largest errors by a large amount and improving the number of RTK fixes makes combination mode masking a beneficial improvement method for RTKLIB type of GNSS processing software.

The masking decreased RTKLIB 3D RMS error on all masking modes except DSM based masking without the multipath interference masking. Combined masking decreased the 3D RMS error of the RTKLIB solutions by 35 % from 6.05 m to 3.95 m while it also improved the distribution of the positioning errors. In addition to accuracy, fix availability also improved from 37 % to 49 % totaling an increase of 32 %, while the number of correct fixes increased by 28 %. The increase in positioning performance was very significant for RTKLIB processing.

Image based mode was the second best on all metrics being somewhat less efficient in RTKLIB processing than the combined masking method, decreasing RMS error by 28 % and improving fix percentage by 27 %. The 3D model based methods were less efficient than the image based mode; the multipath interference masking mode providing better performance than the mode without multipath interference masking. Both DSM based masking methods improved fix percentage by 22 %. Multipath interference enabled masking decreased 3D RMS position error by 22 % while the mode without multipath interference masking increased the RMS error by 2 %.

The improvement in RTK performance is beneficial for RTKLIB type processing software for all used masking methods except for 3D model based masking without multipath interference masking, which provides fairly similar results to the unmasked mode. From these results it can be inferred that the signal masking may help to mitigate multipath interference related errors in addition to NLOS reception related errors despite the receiver implementing a multipath mitigation technique such as APME+. The mitigation of multipath interference errors can be observed in the better performance of DSM with multipath interference masking compared to the DSM without said masking. As the masking modes with and without multipath

interference masking operate otherwise identically, the effect is down to the masking of potential multipath interference areas.

In RTKLIB processing all metrics improved for all test modes when compared to positioning without masking, except the RMS positioning errors for 3D model based masking without multipath interference masking. The increase in RMS error was down to a particular event in which the test vehicle was standing still for a few minutes in traffic lights in a heavily obstructed location on the morning urban route causing very large errors to the PVT solution. The solution had at first considerable errors which caused following obstruction masks to be computed in wrong locations causing even larger errors due to erroneous mask location with the effect propagating from epoch to epoch until the software was able get a new good quality solution. Other test modes are however able to process these same epochs with much better results. This kind of error propagation in a masked solution is a problem for 3D model based masking methods as the masking requires an initial location estimate and the errors in this initial estimate will cause additional errors in subsequent solutions. Image based and combined masking methods will overcome these problems with the use of image based masks which are referenced with timestamps instead of location estimates. 3D model based masking with multipath interference masking was able to avoid as large errors with the masking of potential multipath directions.

From the Septentrio PPSDK processing results we can observe that no masking mode provided best results overall in all metrics; the fix percentages and 3D RMS errors indicate different modes performing well. Fix percentage was highest for unmasked solution but 3D RMS error was smallest for combined masking mode. Combined masking decreased RMS error by 9 % while fix percent and correct fix percent were decreased by 4 % and 3 %, respectively. Image based and 3D model without multipath masking modes provided results from between combined and no masking modes. Clearly the worst masking mode for Septentrio processing was DSM based masking with multipath interference masking causing significant degradation of the positioning performance.

Signal filtering algorithms in Septentrio processing software were most likely able to mitigate multipath interference more effectively than NLOS reception due to multipath interference being easier to detect using the information available for the processing software. If multipath interference errors are already compensated by the Septentrio algorithms before ISMASK filtering, filtering out multipath interference satellites will also remove potentially beneficial satellites from the PVT computation degrading the quality of the resulting solution. As such the multipath interference area masking is the most inadvisable of the studied techniques for this type of processing software. However, if there is not this kind of a satellite selection implementation in the processing software—as is the case with RTKLIB—there is potential also for multipath interference masking.

The distribution of positioning errors for the PPSDK solution, which can be seen from the histogram in figure 25, was slightly better for the unmasked solution than the solution obtained with combined masking mode despite RMS error being smaller for the combined mode solution. In the light of the distribution of the errors the solution accuracies may be regarded fairly similar regardless the masking mode.

For Septentrio software the 3D model based masking without multipath areas offers a minor improvement in the positioning accuracy while managing to keep the correct fix percentage the same as in unmasked solution but having a small reduction in fix availability. The image based masking method provides a small decrease in performance on all used metrics. The combined masking mode is here seen as the most effective mode as it provides the largest increase in positioning accuracy despite the decrease in fix percentage as no method is able to improve or even provide equal fix availability level as the unmasked method.

The following chapters will present how the ISMASK masking affects RTK positioning performance in different environments. From the results presented here it is determined that the combined masking mode is the optimal and the most efficient masking mode overall for both processing software packages. Consequently, the main focus in following sections will be on the effects of combined masking for the RTK performance and all presented results will be from combined masking unless otherwise stated.

4.2 Highway route

The highway is an environment where the obstructions are at their smallest. The biggest problems in terms of obstructions are underpasses and bridges going overhead. There was also some amount of foliage and buildings causing obstructions but they were either only in specific locations or located far away from the road. The lack of obstructions in the highway environment means that there is less potential for multipath interference and especially NLOS reception, diminishing the potential benefits of satellite masking.

The results of the highway datasets are presented in tables 3 and 4. As we can see from the tables, the fix availabilities were on a considerably higher level in the highway environment than in all datasets on average. The positioning errors were also below all test average level. The good positioning performance is due to the relatively unobstructed environment.

Table 3: RTKLIB results of highway datasets

	No Mask	Model	Model No Multipath	Image	Combined
3D RMS Error [m]	1.69	1.7 (+1%)	1.68 (-1%)	1.77 (+5%)	1.77 (+5%)
Fix %	74 %	76 % (+3%)	75 % (+1%)	81 % (+9%)	82 % (+11%)
Correct Fix %	74 %	76 % (+3%)	75 % (+1%)	81 % (+9%)	82 % (+11%)

Table 4: Septentrio PPSDK results of highway datasets

	No Mask	Model	Model No Multipath	Image	Combined
3D RMS Error [m]	1.12	1.36 (+21%)	1.14 (+2%)	0.86 (-23%)	0.89 (-21%)
Fix %	89 %	88 % (-1%)	89 % (0%)	89 % (0%)	88 % (-1%)
Correct Fix %	89 %	88 % (-1%)	89 % (0%)	89 % (0%)	88 % (-1%)

Despite the lack of obstructions, the ISMASK system was able to improve fix availability of RTKLIB processed solutions by 11 % or alternatively decrease number of non-fixed solutions by approximately 30 %. The combined masking had, however, negative effect on 3D RMS error increasing the RMS error by 5 %. The performance of image masking was similar to combined mode while DSM modes were significantly less effective in improving fix availability while providing smaller improvement in positioning accuracy.

For Septentrio PPSDK solutions the combined masking mode had a substantial decrease of 20 % for the RMS error while decreasing also the fix percentage by 1 %. Image based masking performed in fact better than the combined masking in the highway environment. DSM based masking modes did not improve Septentrio solutions in the highway environment. The masking system can be seen to perform better for Septentrio processing software in the highway environment than in more obstructed environments.

The 3D RMS errors for all test modes were quite substantial for RTK positioning in an unobstructed environment. The errors are mostly due to the few obstructions on the route, mostly bridges blocking satellite visibility causing cycle slips, and underpasses—at the both ends of the route—causing loss of solution and poor quality solutions right after the solution has been reacquired. Bad quality solutions were located mostly after bridges and the worst quality solutions were in underpasses. The ISMASK system was typically able to improve positioning performance by decreasing the time to resolve integer ambiguities and get a fixed solution and thereby resulting in an improved fix availability. All epochs in the highway datasets had a reference solution with acceptable quality estimators.

In the highway datasets there was a large difference in RTKLIB solution quality between the morning and afternoon datasets. The decrease in quality is clear from the statistics of the unmasked solution: RMS error grew from 1.25 m in the morning to 2.02 m in the afternoon while fix percentage fell from 87 % to 62 % in the morning and afternoon, respectively. These differences are related to a drop in the number of available Galileo satellites between the test drives which can be observed in figure 21. There were, however, differences also in Septentrio PPSDK solutions between the morning and afternoon datasets despite the processing software using only GPS and GLONASS satellites in RTK mode.

A histogram of the RTKLIB positioning errors for morning and afternoon datasets is presented in figure 26. As can be seen from the figure, most of the epochs in the morning dataset were good quality solutions but in the afternoon dataset a much larger number of the epochs had solutions with larger position errors. The average number of satellites in RTKLIB PVT solutions in the morning dataset was 17 including GLONASS satellites for which the integer ambiguities could not be fixed but the average number of satellites dropping to 13 in the afternoon dataset.

The difference in satellite availability affected strongly the potential of the ISMASK system. For the morning dataset, the ISMASK system was able to slightly improve the RTKLIB positioning performance as the fix percentage increased from 87 % to 91 % (87 % to 90 % for correct fixes) although the RMS error remained unchanged in the morning data. However, in the afternoon dataset the ISMASK masking was able to

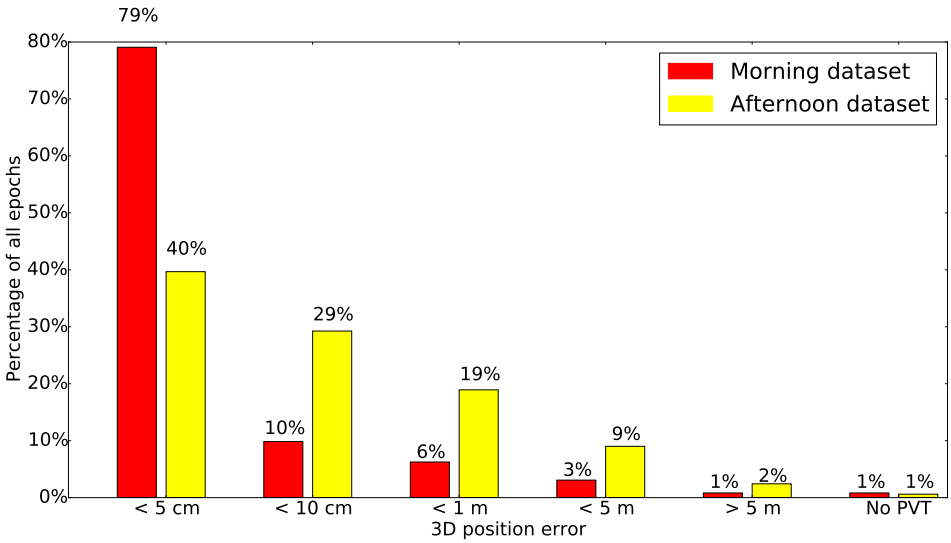

Figure 26: Histogram of RTKLIB position errors on the highway route

improve RTKLIB's fix availability more significantly. The combined masking method improved fix percentage from 62 % to 73 % which is the same as the improvement in correct fix percentage while RMS error increased by 7 % compared to the unmasked mode in the afternoon highway dataset.

Septentrio PPSDK solutions were improved in terms of accuracy much more—by combined masking mode—in the afternoon dataset than in the morning dataset. The computed RMS error decreased by 4 % and 32 % for morning and afternoon datasets, respectively. The fix percentages remained however the same for masked and unmasked solutions.

From the highway datasets it can be concluded that the ISMASK masking system can improve positioning even in environments such as urban highway where most of the obstructions are occasional bridges and underpasses. The performance improvement in RTKLIB type software is achieved in fix availability while the improvement in Septentrio type software is achieved in positioning accuracy. The potential increase in RTKLIB positioning performance in such environment seems to be larger if the amount of available satellites is lower, the effect possibly being due to the larger effect of a single bad signal in a case where the total number of satellites is low. Septentrio PPSDK results seem to vary also depending on the satellite geometry but to lesser extent than RTKLIB results.

4.3 Urban route

The urban route runs in shallow urban canyons in areas with wide streets. The GNSS positioning environment is much harder than the one on the highway route. The obstructions comprise of multi-story buildings and other structures common to urban areas. In an urban environment the potential sources for multipath interference and NLOS reception are much more common than in open highway areas and consequently

there are more satellites to be filtered out by the ISMASK system.

The results obtained from the urban datasets are presented in tables 5 and 6. The same general observations can be made from the urban datasets as on other presented results: Septentrio PPSDK was able to perform better than RTKLIB, masking improved the results for RTKLIB considerably more than for Septentrio PPSDK, the combined masking method provided best masking results, and the DSM based method with multipath interference masking provided the worst results for Septentrio PPSDK. Due to the occasionally challenging environment, 3 % of the epochs do not have reference data of required quality. The lack of reference data causes data from the related epochs to be discarded from reference related statistics.

Table 5: RTKLIB results of urban datasets

	No Mask	Model	Model No Multipath	Image	Combined
3D RMS Error [m]	6.78	4.06 (-40%)	7.65 (+13%)	3.39 (-50%)	3.22 (-53%)
Fix %	36 %	51 % (+42%)	49 % (+36%)	53 % (+47%)	56 % (+56%)
Correct Fix %	35 %	49 % (+40%)	47 % (+34%)	52 % (+49%)	54 % (+54%)

Table 6: Septentrio PPSDK results of urban datasets

	No Mask	Model	Model No Multipath	Image	Combined
3D RMS Error [m]	1.15	1.23 (+7%)	1.13 (-2%)	1.22 (+6%)	1.05 (-9%)
Fix %	85 %	80 % (-6%)	85 % (0%)	83 % (-2%)	83 % (-2%)
Correct Fix %	84 %	79 % (-6%)	84 % (0%)	82 % (-2%)	82 % (-2%)

The combined masking mode improved RTKLIB fix percentage by over 50 %, also for the correct fixes. RMS position error was also decreased to less than half of the original error. The overall increase in RTK positioning performance was very substantial. Other masking methods provided lesser performance improvement but were still able to improve the results on a meaningful level.

The obtained results for combined masking in Septentrio PPSDK provided similar performance as in the data from all datasets; RMS error decreased by 9 % but at the cost of a reduction in fix availability. The 3D model based masking without interference area masking performed quite identically to the unmasked solution offering only a little improvement.

Maps presenting the change in RTKLIB positioning errors between the unmasked solution and the solution obtained by combined camera and image masking are illustrated in figures 27 and 28 for morning and afternoon datasets, respectively. As can be seen from the maps, the effect of signal masking was negative on some epochs, positive on some, and on many epochs the effect was less than 10 cm. The limit of 10 cm errors was chosen for the sake of clearer presentation due to mitigation of small magnitude noise effects and to highlight the areas where the positioning accuracy undergoes the largest changes. The images do not provide a clear indication on what kind of locations the system is beneficial and where not. The effects are obviously dependent on the location of satellites causing differences between the

morning and afternoon datasets. The solution quality has however improved on some locations in both datasets, many of these locations being quite obstructed or right after obstructed areas. The characteristics of performance alterations seem nevertheless relatively random.

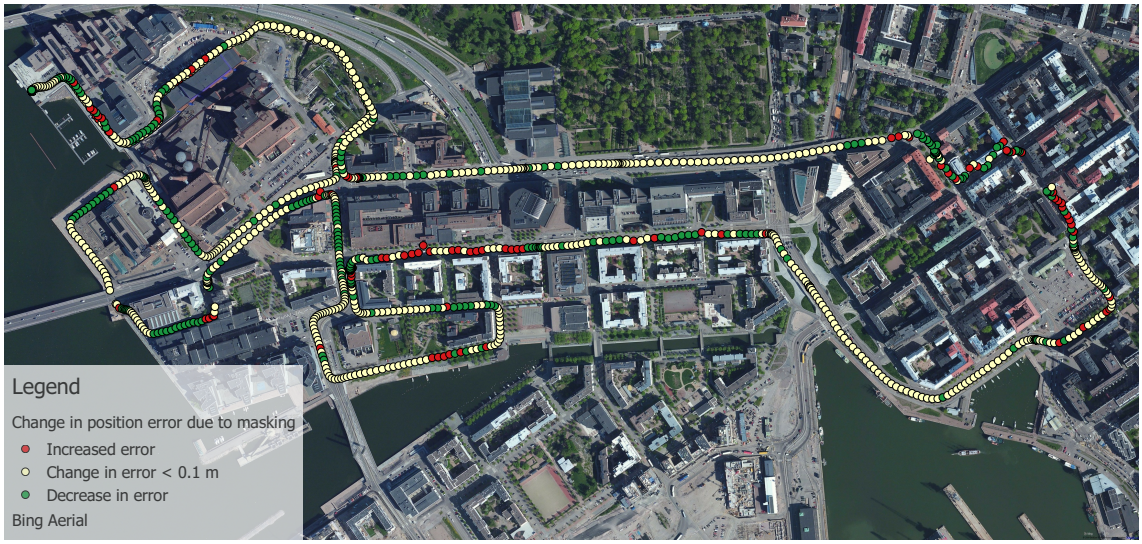

Figure 27: Change in RTKLIB position errors due to masking on the morning dataset

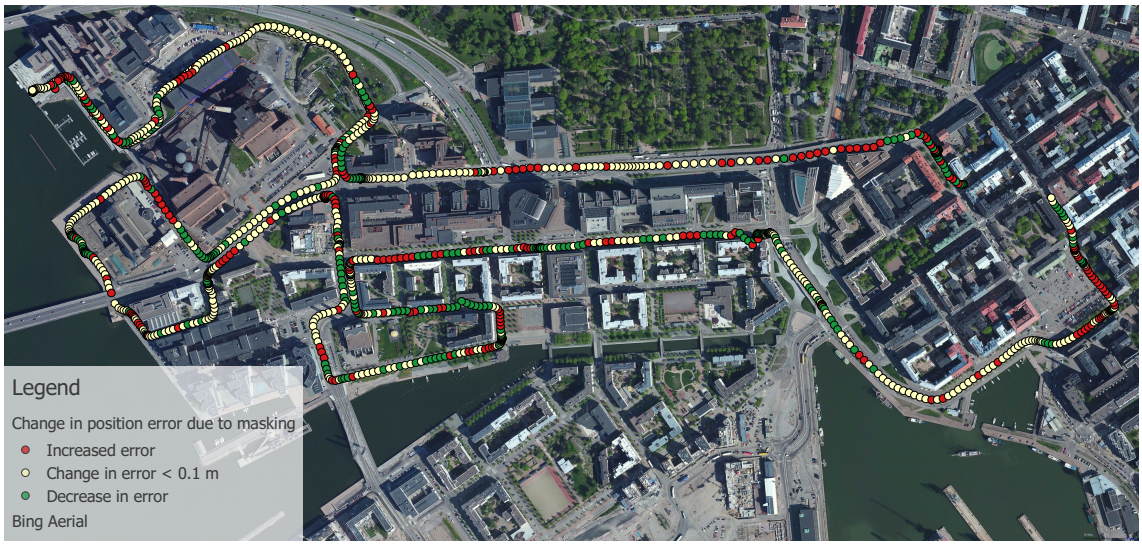

Figure 28: Change in RTKLIB position errors due to masking on the afternoon dataset

A similar difference in the number of available satellites between morning and afternoon datasets as in the highway datasets was present also in the urban test data. Average number of satellites in RTKLIB solutions decreased from 13 in the morning dataset to 9 in the afternoon dataset including GLONASS satellites for which

ambiguity resolution was disabled. Both negative and positive effects in positioning accuracy were more common in the afternoon as can be seen from the position error maps in figures 27 and 28.

For RTKLIB solutions combined masking mode was able to reduce RMS position error by 30 % in the morning dataset but in the afternoon the reduction was as large as 69 %. Similar to the positioning accuracy, the difference in fix percentages was considerable. Combined masking was able to improve fix percentage from 50 % in the unmasked solution to 71 % in the morning dataset and from 24 % to 43 % in the afternoon dataset. The increase of 79 % in fix percentage in the afternoon dataset is a very substantial improvement, while the 42 % increase in the morning dataset is also significant. The masking seems to increase positioning performance more at times when the number of used satellites is lower similarly to the results obtained from highway datasets.

Results for Septentrio PPSDK between morning and afternoon datasets were very different to RTKLIB results. RMS error was decreased by 11 % in the morning dataset while providing the same level of increase in the RMS error in the afternoon dataset. The accuracy of the afternoon dataset was, however, much greater than the accuracy of the morning dataset; unmasked RMS error was 1.58 m in the morning and 0.52 m in the afternoon. The greater accuracy seems to diminish the potential of the ISMASK system. The fix percentages between morning and afternoon datasets underwent equal changes due to combined satellite masking.

The ISMASK processing system improves RTK positioning performance in the urban environment more than in the highway environment when using RTKLIB type of processing software. For Septentrio PPSDK type software the effects in the urban environment are similar to results obtained from all datasets providing moderate RTK performance improvement. The filtering system is able to provide most substantial improvements at times when there are a limited number of satellites available or the quality of solutions are lower. The improvements in RTKLIB type processing are significant however also when having a greater number of available satellites and with higher quality solutions.

4.4 Dense urban route

The harshest satellite navigation environment in the tests was on the dense urban route. The route runs in deep urban canyons formed of multi-story buildings on the sides of quite narrow urban roads. The combination of buildings and narrow streets makes the obstructions quite substantial despite the majority of the buildings being less than 10 floors high. The canyons provide a lot of potential for multipath interference and NLOS propagation [27].

The results of the dense urban tests are presented in tables 7 and 8. The RTK positioning environment was clearly the most difficult as the fix percentages were on much lower level for both processing software packages than on the other routes. The differences between RTKLIB processing performances of test modes were similar to the results of all datasets, except that the DSM without multipath interference masking performed better than the image based masking. As with other routes, the

combined masking provided best results for RTKLIB processing. Masking was not able to improve Septentrio PPSDK positioning accuracy in the same level as in less obstructed environments.

Table 7: RTKLIB results of dense urban datasets

	No Mask	Model	Model No Multipath	Image	Combined
3D RMS Error [m]	7.73	7.46 (-3%)	6.65 (-14%)	7.08 (-8%)	6.31 (-18%)
Fix %	16 %	21 % (+31%)	22 % (+38%)	21 % (+31%)	24 % (+50%)
Correct Fix %	13 %	16 % (+23%)	18 % (+38%)	16 % (+23%)	18 % (+38%)

Table 8: Septentrio PPSDK results of dense urban datasets

	No Mask	Model	Model No Multipath	Image	Combined
3D RMS Error [m]	2.08	3.31 (+59%)	2.04 (-2%)	2.2 (+6%)	1.95 (-6%)
Fix %	43 %	37 % (-14%)	40 % (-7%)	39 % (-9%)	39 % (-9%)
Correct Fix %	37 %	31 % (-16%)	37 % (0%)	36 % (-3%)	34 % (-8%)

The dense urban route started and ended in an area with good satellite visibility and minimal obstructions to provide good quality solutions to both ends of the route for data quality monitoring purposes; however, these epochs were not taken into account in the analysis as these sections are not representative of dense urban environment. A lot of epochs were also discarded from reference solution related analysis due to insufficient precision of the reference solution. The number of rejected reference solution epochs was roughly half of all the epochs.

In RTKLIB processing, the ISMASK system was able to decrease the position RMS error by 18 % from 7.73 m without masking to 6.31 m with combined masking mode. The fix percent was increased from 16 % to 24 % providing an increase of 50 %. For the correct fixes the increase was however only 38 %. As seen in the fix percentages the ISMASK system seems to increase the number of false fixes in dense urban environment in addition to increasing the number of correct fixes. The overall positioning accuracy, however, improved considerably despite the increase in false fixes. The performance improvement with respect to location appeared as random as it was in the case of urban route—the masking might have improved the solution in one epoch and degraded the solution accuracy soon after even while driving the same road to the same direction. The locations in which positioning performance increased are, however, fairly constant between the morning and afternoon datasets. The performance metrics of other masking methods indicate worse performance than combined masking but the masking was still able to improve performance with respect to the unmasked solution for all other masking modes.

Septentrio PPSDK results were similar to other routes in that the RTK positioning performance improved in accuracy for combined masking mode but fix availability decreased mitigating the improvement in RMS error. The reduction in RMS error was 6 % while the fix percentage decreased by 9 %. The performance of 3D model without multipath interference masking was from between the combined masking

and unmasked modes, with small improvement in RMS error and decrease in fix percentage. Image based and 3D model based modes were unable to improve the solutions in any of the used metrics. It is important to note that Septentrio positioning was also able to obtain considerably better results in this kind of environment on all test modes when compared to the RTKLIB position solutions.

Unlike in the other routes, there was not as significant a drop in the number of available GPS and Galileo satellites between morning and afternoon sessions on the dense urban route as on other routes as indicated by figure 21. The average number of satellites in RTKLIB solutions nevertheless decreased from 9 to 8 between the morning and afternoon datasets. On the dense urban route, the masking system was able to improve RTKLIB positioning performance more in the morning dataset unlike on the highway and urban routes. The improvement in performance in terms of accuracy was greater in the morning than in the afternoon also for Septentrio PPSDK solutions.

In the morning RTKLIB dataset the combined masking method decreased RMS position error from 9.43 m to 7.39 m totaling a decrease of 21 %. Fix percentage increased from 14 % to 24 %, totaling an increase of over 70 % which is very significant although the increase of correct fixes was closer to 50 %. However, there were large differences in the obtained results between morning and afternoon sessions. In the afternoon dataset the root mean square error decreased 1 % from 4.46 m to 4.41 m. Afternoon fix percentage increased by 28 % from 18 % to 23 % while the increase of 8 % in correct fixes was considerably less. The overall difference between the morning and afternoon datasets might have been more due to difference in satellite locations than number of available satellites as the difference in number of available satellites was quite small between the morning and afternoon test drives.

Positioning accuracy of Septentrio solutions—characterized by the 3D position RMS errors—improved by 8 % in the morning and by 4 % in the afternoon while the fix percentages decreased by 6 % and 14 % in morning and afternoon, respectively. The ISMASK systems performs quite much better in the morning but the decrease in fix availability was significant also in the morning data diminishing the benefits of improved accuracy.

The performance of the ISMASK system is not as prominent in dense urban environment as in urban environment but the improvement in RTK positioning performance especially when using RTKLIB type processing software is significant. Like in other environments the RTK performance when using Septentrio PPSDK type processing software is closer to unmasked performance but there can, arguably, be seen some benefit in satellite masking. There was a noticeable difference between datasets collected at morning and afternoon which might partially be due to a slightly smaller number of available satellites in the afternoon and partially due to the change in location of satellites between the test drives.

The results from urban datasets should be considered less reliable than the ones from highway and urban datasets as the metrics requiring reference data—3D position RMS errors and correct fix percentages—were missing approximately half of the all epochs and thereby causing bias in the results and less comprehensive coverage of the environment. The fix percentage by contrast was computed for all epochs and

provides more representative results from the dense urban environment.

4.5 Masking with different processing algorithms

The results presented earlier show that the ISMASK system does not improve overall performance of RTK positioning in as high levels when using Septentrio PPSDK as processing software as it does for RTKLIB positioning. There is improvement in positioning accuracy in terms of position RMS errors but the distribution of the errors does not improve. The number of fixed solutions also decreases when using ISMASK satellite masking as is the case with also the number of correctly fixed solutions.

The signal and satellite selection and masking algorithms in Septentrio PPSDK processing seem to work relatively well in removing bad quality signals from the PVT solution computation causing the observed lesser effect of the ISMASK system. A demonstration of this can be seen in figures 29 and 30. Figure 29 presents the obstruction mask generated by the combination method at epoch 301095 SOW (seconds of week) and the sky-view image of the same epoch. The epoch 301095 SOW occurs during the afternoon urban test drive. Note that the image is mirrored and also oriented according to the vehicle direction, not in the same system as the obstruction mask. The epoch along with some time before and after it (a few minutes in total) was spent in traffic lights and during that time the unmasked RTKLIB solution experienced large position errors, up to a few meters in magnitude. Position errors for both unmasked and combination mode solutions computed with RTKLIB and Septentrio PPSDK can be seen in figure 30. The unmasked RTKLIB solution was a float RTK solution while other solutions were fixed RTK solutions.

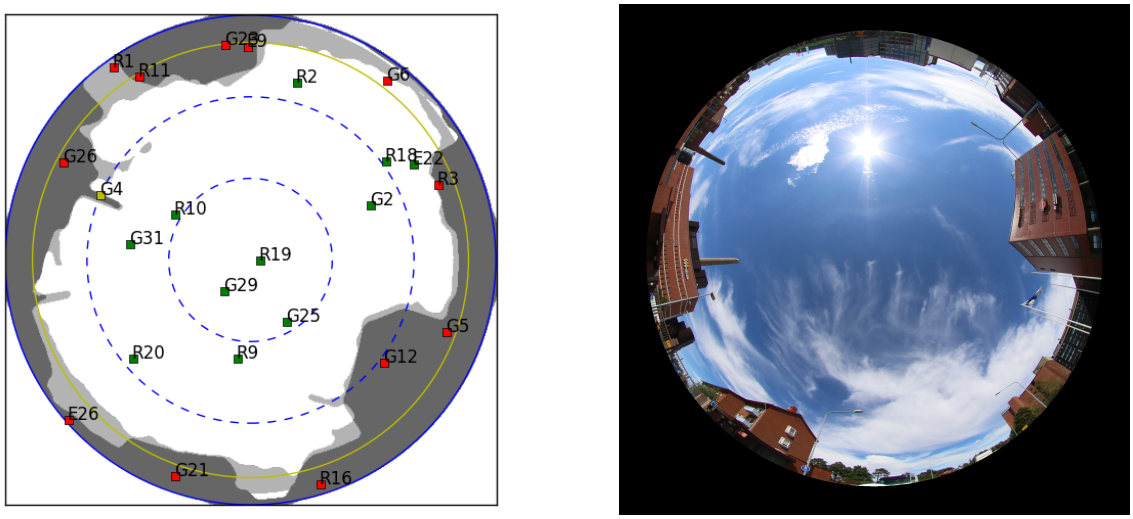

Figure 29: Obstruction mask (left) and sky-view image (right) from afternoon urban dataset

The unmasked RTKLIB solution has very large position errors but the masked RTKLIB solution along with both Septentrio PPSDK solutions have errors of much

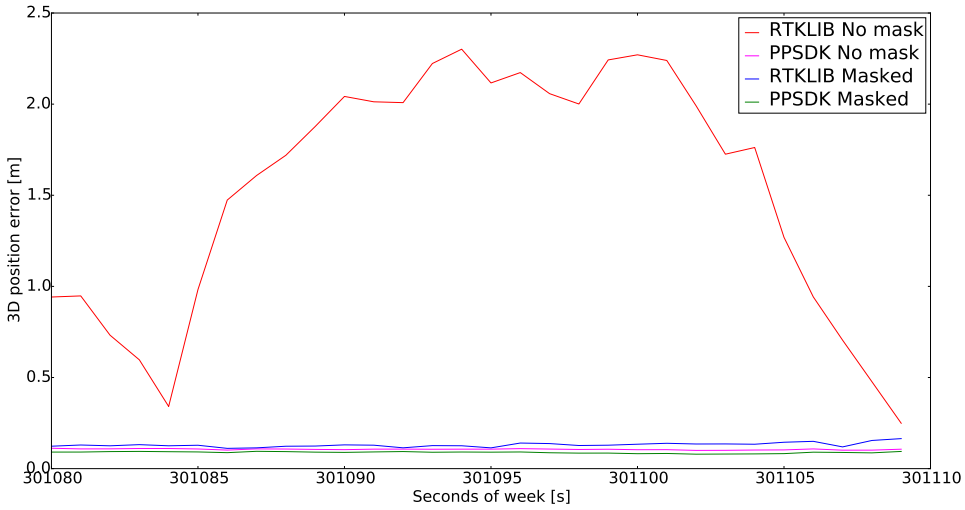

Figure 30: 3D position errors of different solutions around epoch 301095 SOW

smaller magnitude, approximately 10 cm. The large errors in the unmasked RTKLIB solution are caused by NLOS reception. The unmasked RTKLIB solution was using the G26 and E9 satellites which are clearly blocked by obstructions while the R3 and R11 satellites were both obstructed but the received signal might be only refracted instead of reflected as the satellites are very close to obstruction edges. The unmasked Septentrio solution used the R3 and R11 satellites but the processing software had discarded the signals from G26 from PVT computations presumably due to signal selection and filtering algorithms deeming them bad quality signals. E9 satellite was not used by Septentrio PPSDK because the software is unable to use Galileo satellites in RTK positioning. R3 and R11 satellites were both used in the unmasked PPSDK solution without causing large errors which leads one to believe that it was indeed either G26, E9, or both satellites that caused the large errors in the unmasked RTKLIB solution.

In order to assess the individual contributions of NLOS satellites G26 and E9 to the positioning errors, the unmasked RTKLIB data had to be processed while excluding the E9 satellite from processing. 3D positioning errors for unmasked RTKLIB solutions processed with all satellites and all satellites except E9 are presented in figure 31. By examining the RTKLIB results when discarding the E9 satellite from processing of the unmasked solution it can be seen that the majority of the errors is caused by the G26 satellite with E9 having significantly lesser although still damaging impact. It can be seen that Septentrio PPSDK is able to discard erroneous NLOS satellite data without additional signal masking due to Septentrio's satellite selection algorithms.

In this case it can be seen that the ISMASK masking system is able to improve the Septentrio PPSDK solution by a small amount, but the increase in the level of that between the RTKLIB solutions is not possible to obtain. The unmasked Septentrio solution is a good quality solution because it has been able to detect and

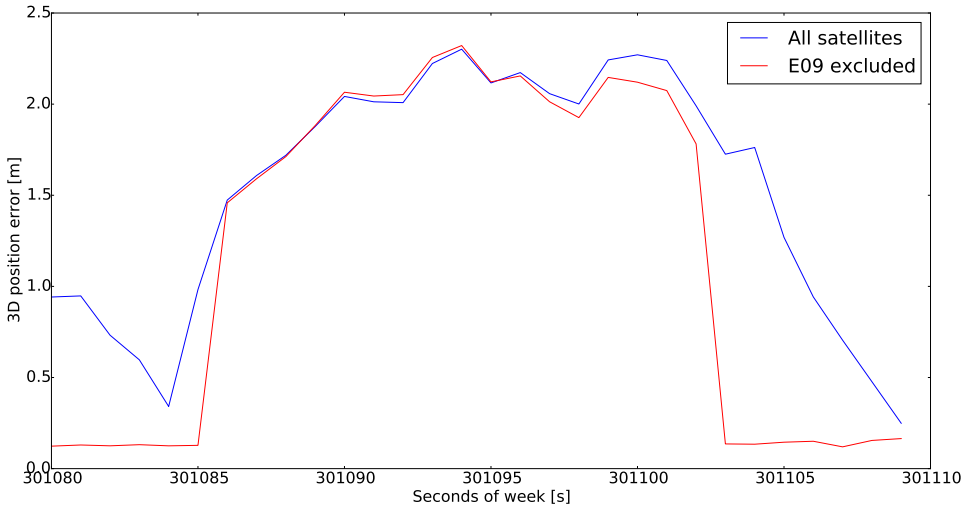

Figure 31: 3D position errors of RTKLIB solutions with different satellites around epoch 301095 SOW

exclude signals from the clearly obstructed satellite.

There is however potential to improve Septentrio satellite selection and filtering algorithms more significantly by utilizing the data provided by the ISMASK system in the selection and filtering algorithms themselves instead of adding an additional layer of filtering. There was unfortunately no way to study this possibility within the context of this work or the ISMASK project.

The ISMASK satellite masking and filtering system is able to improve the Septentrio PPSDK solution significantly in some situations despite the overall effect being close to neutral. A position error plot from one of such occasions is presented in figure 32. The figure contains 3D position error data of 10 epochs before and after epoch 290398 SOW for unmasked and combination mode masked Septentrio PPSDK solutions. The unmasked solution has a high error peak for the epoch in question while the masked solution is able to avoid such large errors. The unmasked solution drops to float mode for a few subsequent epochs during which the masked solution is able to keep RTK in fixed mode. The position errors however return to similar levels soon after the epoch in question.

Obstruction mask (left) and sky-view image (right) of the epoch 290398 SOW are displayed in figure 33. Masked and unmasked solutions have used the same satellites in the PVT solutions except for satellite G19 which is a NLOS satellite and filtered out by the ISMASK system in the masked solution, but not by Septentrio's algorithms. In situations like this there is a clear benefit of using satellite masking even for software using advanced satellite selection and filtering algorithms but the negative effects during other times mitigate the improvement in performance overall.

The results presented in this chapter give a comprehensive view of how does the ISMASK system affect RTK positioning performance and how do the different mask generation methods compare. Conclusions from the results given in this chapter are

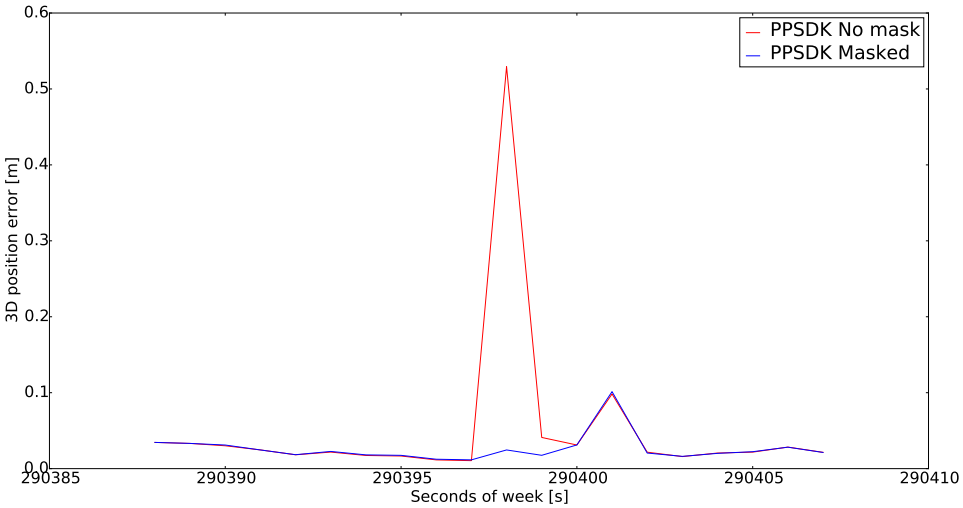

Figure 32: Septentrio PPSDK 3D position errors at one point on the urban route

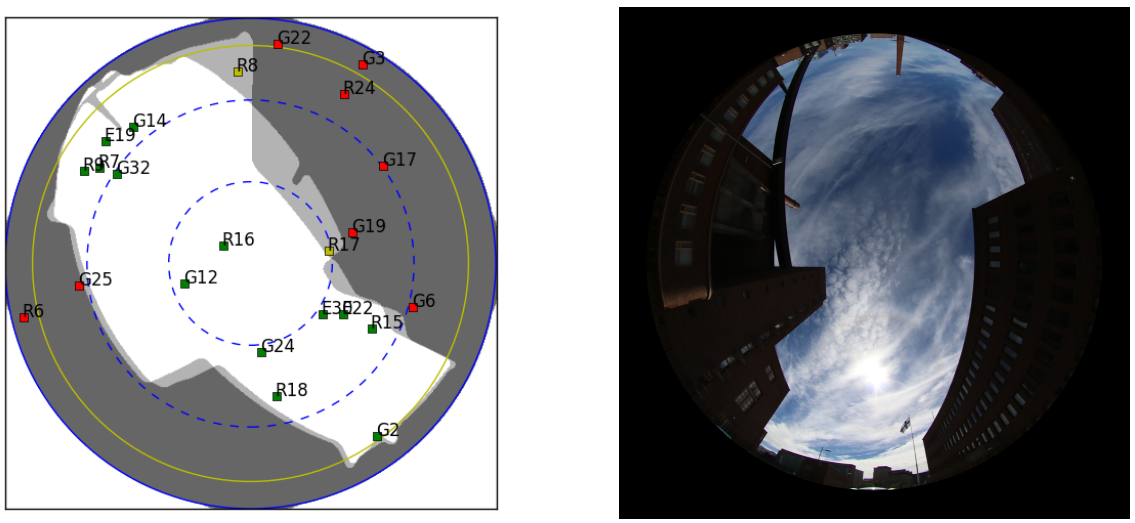

Figure 33: Obstruction mask (left) and sky-view image (right) from morning urban dataset

presented in the next chapter.

5 Conclusions

This work presents a comprehensive assessment of RTK positioning aiding using 3D models and fisheye camera. The developed ISMASK software is a software combining obstruction masking with RTK positioning in a way that allows different kinds of system configurations for research purposes. In order to test the software, a hardware test bench was developed. The test bench was used in the test campaign to record a considerable amount of positioning and aiding data which was then processed using the ISMASK software.

The ISMASK software can be used in five different processing modes:

- no masking, legacy processing
- 3D model based masking
- 3D model based masking with masking of multipath interference areas
- sky-view image based masking
- combination of 3D model and sky-view image based masking.

The different processing modes produce different masking characteristics depending on the obstructions around the receiver. An important part of the work was to determine the best way to generate the obstruction masks out of these four possibilities.

Earlier studies have demonstrated similar satellite masking concepts but often in either non-RTK positioning or with considerably smaller test campaigns. The common aspect in many studies conducted earlier in the subject is that they are more like proof of concept studies than assessments of actual positioning performance. The earlier studies concentrating on at least partially similar systems have not managed to show any significant improvement for RTK positioning performance at least on the scale studied in this work. Exactly similar systems utilizing both camera and 3D models as data source have not been studied before.

The data recorded in the test campaign included various different satellite navigation environments with varying levels of obstructions to assess the effects of masking in general and different masking methods in particular to the performance of RTK positioning. The tests were designed to provide reliable results on urban RTK usage.

Before the processing of test campaign data could be started a pre-processing step was performed during which also the quality of recorded data was investigated. From the information obtained in the pre-processing stage it was concluded that all systems worked according to expectations except that the quality of the reference solution did not reach the desired level in the most demanding navigation environments. Appropriate steps were then followed to isolate the epochs in which the reference solution was not of sufficient quality.

The ISMASK system was used with a commercial GNSS processing software Septentrio PPSDK and with an open-source alternative RTKLIB. There are many differences between the two software packages and consequentially the results of

satellite masking were different for the two processing software packages. The most important characteristics of RTK positioning performance for the two software from all recorded data being:

- Septentrio PPSDK fix availability decreases
- RTKLIB fix availability increases
- Septentrio PPSDK 3D RMS error decreases
- RTKLIB 3D RMS error significantly decreases
- Septentrio PPSDK error distribution remains the same
- RTKLIB error distribution improves
- the largest improvement in performance for RTKLIB was observed in shallow urban environments and second largest in dense urban environments
- the improvement in Septentrio PPSDK performance is greater the less obstructions there are.

The first four of these influences are shown in more detail in table 9 which summarizes the overall effects of combination mode masking compared to unmasked PVT solutions. The table presents 3D position RMS errors, percentage of fixed solutions out of all solutions, and percentage of correctly fixed solutions out of all solutions.

Table 9: Summary of effects of combined masking

	RTKLIB		Septentrio PPSDK	
	No Mask	Combined masking	No Mask	Combined masking
Position 3D RMS Error [m]	6.05	3.95 (-35%)	1.43	1.3 (-9%)
Fix %	37 %	49 % (+32%)	70 %	67 % (-4%)
Correct Fix %	36 %	46 % (+28%)	67 %	65 % (-3%)

The different masking modes provided different results compared to the unmasked solution, with the best results—the ones given above—being from combined image and DSM without multipath interference masking mode. It was concluded from the results that the optimal way to generate the obstruction masks with the ISMASK system is to use both images and 3D models. The overall benefits of the rest of the test modes vary between the two used software packages. Image based masking was the second best mode for RTKLIB processing and 3D model with multipath interference masking provided third best results. All three modes improved the solutions while the 3D model without multipath interference masking provided results similar to unmasked solution. The second best masking method for Septentrio PPSDK was the 3D model without multipath interference masking while the image derived method offered a third best performance. The DSM with multipath interference masking

mode was very unsuitable for Septentrio PPSDK offering the worst results with significant decrease in performance compared to other masking methods and also compared to the unmasked solution.

The ISMASK system clearly improves RTKLIB solutions by a significant amount in both fix availability and positioning accuracy. For commercial Septentrio PPSDK software there is a small decrease in the availability of fixes while the positioning accuracy is improved by a moderate amount. The results are, however, not as clear cut for all test modes as they are for combination masking. All masking modes reduce the fix availability for Septentrio PPSDK, while the fix availability increases considerably for all modes when using RTKLIB for the post-processing. The results for modes other than combined masking are generally somewhere in between the results of combined masking and unmasked solution.

From the results presented in table 9 it can be seen that the unmasked solutions of Septentrio PPSDK are much more accurate than the corresponding RTKLIB solutions. A large contribution to the better accuracy is that PPSDK is able to fix GLONASS integer ambiguities, despite being unable to use Galileo in RTK positioning. The better accuracy is also probably partially down to Septentrio's signal and satellite selection algorithms and partially due to more optimized or more advanced positioning algorithms. The better accuracy in unmasked solutions reduces the potential improvement which can be obtained by the ISMASK masking and filtering system.

In addition to the processing software, the performance of the ISMASK system also differs with relation to the test environments. The effect of different environments is dissimilar between the two processing software packages. The masking provided better results for Septentrio software in environments with fewer obstructions; results were best on highway, worst on the dense urban route, and fall between the two on the urban route. For RTKLIB the masking provided better results in obstructed environments with best results being from urban route, second best from dense urban route, and the worst from highway.

Differences between how the used RTK processing software compute position solutions result in different characteristics in the masking performance. Septentrio PPSDK has more available data due to Septentrio SBF-file format containing additional data compared to the data included in RINEX files used by RTKLIB. How the Septentrio positioning algorithms exactly work is not known due to the closed nature of commercial software. It can however be assumed that there is some sort of signal and satellite selection algorithms which exclude unwanted satellites and possibly compensate for multipath interference. These algorithms diminish the domain in which the ISMASK system operates.

The ISMASK system is an additional filtering layer on top of Septentrio's algorithms potentially disrupting Septentrio's satellite selection algorithms and therefore resulting in a reduction of fix availabilities. Multipath interference correction for satellites for which both NLOS and LOS signals are observed is a likely explanation for why the multipath interference masking performs extremely poorly with Septentrio PPSDK but provides much better results with RTKLIB. The multipath corrections are possible due to APME+ technology used in the receivers. The correc-

tions applied by APME+ in the code and carrier tracking stages are present in both SBF and RINEX files and therefore taken advantage of in both processing software; however, multipath magnitude estimates themselves are not available in RINEX files and thereby not usable in RTKLIB processing. It can be concluded that receiver made compensations—at the levels used in the receivers in question—to multipath interference do not diminish the potential of multipath interference masking, but processing algorithms further utilizing obtained multipath estimates dilute the potential improvement.

The results from using the developed satellite masking system depend heavily on the used test environment. The reasons for PPSDK obtaining better results in open environments than in obstructed may only be hypothesized, one potential cause being more aggressive—due to more obstructions on the masks—filtering interfering with Septentrio's positioning algorithms. It will however remain unknown how the satellite masking would affect the GNSS processing if the information obtained by the ISMASK system would be incorporated into existing masking and filtering algorithms instead of adding another filtering layer.

For RTKLIB solutions, the ISMASK system on the other hand provides the smallest improvement—in fact a decrease in accuracy—in least obstructed environments. This is probably due to simply smallest amount of potential objects to be masked in the unobstructed environments as the ISMASK system acts as a sole filtering layer for RTKLIB and hence does not interfere with existing algorithms. The difference between results from urban and dense urban environments might indicate that the ISMASK system is more suitable for shallower urban canyons and there might be a need for more optimization for the algorithms in more difficult positioning environments.

There was a large drop in the number of available Galileo satellites between the highway datasets recorded in the morning and afternoon. During both sessions the number of GPS satellites remains stagnant. As the RTKLIB processing operates with GPS, GLONASS, and Galileo satellites, but with possibility to fix integer ambiguities to only GPS and Galileo satellites, the smaller number of fixable satellites alters the positioning environment considerably. In the morning there are five available Galileo satellites of which three were on favorable elevations, all being over 20 degrees above horizon, in contrast to 2 available satellites of which 1 is barely above 10-degree elevation mask. This difference in Galileo constellation creates a very different positioning environment between the datasets.

On the highway route RTKLIB positioning accuracy remained the same for masked and unmasked solutions for the morning session and the fix availability improved by a small amount. In the afternoon the positioning accuracy however decreased by a small amount compared to the unmasked solution while fix availability improved considerably. The ISMASK system seems to benefit in open environment from additional Galileo satellites in terms of positioning accuracy. The increase in the number of fixes is more related to the much lower starting level of the unmasked solution—62 % in the afternoon compared to the 87 % in the morning.

In urban datasets there is a similar, but smaller, difference in the number of Galileo satellites but the results are contrary indicating that the added number of

Galileo satellites does not improve the potential of the ISMASK system in more obstructed environments. The reason for this difference in the two environments may also be related to the location of used satellites in addition to just the number of used satellites as the different obstruction environments cause different satellites to be visible on different routes despite satellite locations remaining the same.

The results obtained in the ISMASK project and presented here are designed to be as reliable as possible; however, there are some major considerations to be noted relating to the trustworthiness of the results. The biggest problem being lack of full reference solution coverage which resulted in some of the metrics used in the analysis having to discard the most difficult navigation environments, at least partially. The missing reference data was taken into account in analysis of the results in an attempt to mitigate the damage resulting from the discarded data. The main effect of missing reference data to the results presented here is hence that the results do not represent the most difficult dense urban environments as well as they do other environments. The test campaign despite containing multiple hours of data could still also include a lot more data to improve reliability of the results but due to restrictions set by the project this could not be done. The amount of data would especially improve knowledge of how do the number of available satellites affect the impact of satellite masking.

At the moment of this work the Galileo system had not yet reached its full extent, in fact the initial services were not even launched during the test campaign. Moreover, many commercial RTK software packages do not yet fully support Galileo, including Septentrio PPSDK which was used in this work. The number of available satellites was not at the levels it will reach in the future leaving little room for additional satellite filtering—especially in the dense urban areas. In the near future, when there will be more satellites in the sky, the results of this kind of masking could be more significant in general and especially in urban canyons.

A problem in this work with the Septentrio PPSDK software was that it could not be known how the software's own signal exclusion algorithms work, and therefore signal filtering algorithms could not be optimized to work with them. There is a possibility that a signal masking technique like ISMASK could be integrated with other signal exclusion algorithms in a way in which they would support each other allowing potentially much better results. This would require further development with the full understanding of a navigation software's legacy signal exclusion algorithms, and possibly modifications also to the internal acquisition and tracking loops.

During the various project phases there are many possible missteps which would lead to reduced reliability of the results. A rigorous attention was therefore paid during the project to control as many factors as possible to reveal possible mistakes and to prevent any mishaps. All encountered problems and mistakes were corrected without causing any problems—with exception being the things mentioned here—that could not be fixed and therefore affect the quality of the results.

Based on the test conducted in the Intelligent Signal Masking for GNSS RTK project the ISMASK masking system is able to improve RTK positioning in terms of accuracy in many different environments. The system is able to achieve greater improvement in results when used with less optimized positioning algorithms or

positioning algorithms with less advanced features. The RTK positioning performance in terms of fix availability increases when processed with more general positioning algorithms but decreases when operating with advanced and optimized manufacturer algorithms. There is no reason to assume that the system would not have possibility to achieve even better results with additional optimizations to the algorithms used in this project.

The developed ISMASK system could be commercially viable with some additional development on some receiver configurations even at its current state. However, at least in receivers or software utilizing similar processing to Septentrio PPSDK there would likely be a considerable amount of work needed to integrate the systems. At its current state the software would require mostly optimizations to improve masking performance in some environments and to improve real time capabilities of the software. The ISMASK system has demonstrated significant potential in this type of satellite masking with possibilities for commercial implementation.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[2] European Space Agency. ESTEC. Cited: 18.1.2017, available: http://www.esa.int/About_Us/ESTEC.

[3] J Amt and J Raquet. Positioning for range-based land navigation systems using surface topography. In *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS*, pages 1494–1505. ION, September 2006.

[4] D. Attia, C. Meurie, Y. Ruichek, J. Marais, and A. Flancquart. Image analysis based real time detection of satellites reception state. In *2010 13th International IEEE Conference on Intelligent Transportation Systems - (ITSC 2010)*, pages 1651–1656. IEEE, 2010.

[5] David Betaille, Stéphan Miquel, F Godan, and François Peyret. 3D-city-model-aided GNSS accurate positioning with integrity provision using simplified geometry of buildings. In *European Navigation Conference 2015*, 2015.

[6] Basudeb Bhatta. *Global Navigation Satellite Systems: Insights into GPS, GLONASS, Galileo, Compass, and Others*. BS Publications, 2010.

[7] Paul Blunt. GNSS Signal Acquisition and Tracking. In Scott Gleason and Demoz Gebre-Egziabher, editors, *GNSS Applications and Methods*, chapter 2, pages 23–54. Artech House, 2009.

[8] Michael S. Braasch. Multipath effects. In Bradford W. Parkinson and James J. Spilker, editors, *The Global Positioning System: Theory and Applications*, volume I, chapter 14, pages 547–568. American Institute of Aeronautics and Astronautics, 1996.

[9] J Bradbury, M Ziebart, PA Cross, P Boulton, and A Read. Code multipath modelling in the urban environment using large virtual reality city models: Determining the local environment. *Journal of Navigation*, 60(01):95–105, 2007.

[10] P. Brocard, D. Salos, O. Julien, and M. Mabilleau. Performance evaluation of multipath mitigation techniques for critical urban applications based on a land mobile satellite channel model. In *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, pages 612–625. ION, May 2014.

[11] Grover R. Brown. Receiver Autonomous Integrity Monitoring. In Bradford W. Parkinson and James J. Spilker, editors, *The Global Positioning System: Theory and Applications*, volume II, chapter 5, pages 143–165. American Institute of Aeronautics and Astronautics, 1996.

[12] Cindy Cappelle, Maan El Badaoui El Najjar, Denis Pomorski, and François Charpillet. Localisation in urban environment using GPS and INS aided by monocular vision system and 3D geographical model. In *2007 IEEE Intelligent Vehicles Symposium*, pages 811–816. IEEE, 2007.

[13] Open Geospatial Consortium. OGC City Geography Markup Language (CityGML) Encoding Standard. Standard, Open Geospatial Consortium, 2012.

[14] van AJ Dierendonck, Pat Fenton, and Tom Ford. Theory and performance of narrow correlator spacing in a GPS receiver. *Navigation*, 39(3):265–283, 1992.

[15] Rudy Ercek, Philippe De Doncker, and Francis Grenez. Statistical determination of the PR error due to NLOS-Multipath in Urban Canyons. In *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS*, pages 1771–1777. ION, September 2006.

[16] European Global Navigation Satellite Systems Agency. Galileo begins delivery of initial services, December 2016. Press release GSA/PR/16/08.

[17] P. Franois, B. David, and M. Florian. Non-line-of-sight GNSS signal detection using an on-board 3D model of buildings. In *2011 11th International Conference on ITS Telecommunications (ITST)*, pages 280–286. IEEE, August 2011.

[18] Demoz Gebre-Egziabher, Mark Petovello, and David Bevly. Integration of GNSS and INS: Part 1. In Scott Gleason and Demoz Gebre-Egziabher, editors, *GNSS Applications and Methods*, chapter 6, pages 149–176. Artech House, 2009.

[19] Scott Gleason and Demoz Gebre-Egziabher. Global Navigation Satellite Systems: Present and Future. In Scott Gleason and Demoz Gebre-Egziabher, editors, *GNSS Applications and Methods*, chapter 1, pages 1–22. Artech House, 2009.

[20] Scott Gleason and Demoz Gebre-Egziabher. *GNSS Applications and Methods*. Artech House, 2009.

[21] Scott Gleason and Demoz Gebre-Egziabher. GNSS Navigation: Estimating Position, Velocity, and Time. In Scott Gleason and Demoz Gebre-Egziabher, editors, *GNSS Applications and Methods*, chapter 3, pages 55–86. Artech House, 2009.

[22] Mohinder S Grewal, Lawrence R Weill, and Angus P Andrews. *Global Navigation Satellite Systems, Inertial Navigation, and Integration*. John Wiley & Sons, 2013.

[23] P. D. Groves, Z. Jiang, L. Wang, and M. Ziebart. Intelligent urban positioning, shadow matching and non-line-of-sight signal detection. In *Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing, (NAVITEC), 2012 6th ESA Workshop on GNSS Signals and Signal Processing*, pages 1–8. IEEE, Dec 2012.

[24] Paul D Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech house, 2008.

[25] Paul D Groves. Shadow matching: A new GNSS positioning technique for urban canyons. *Journal of Navigation*, 64(03):417–430, 2011.

[26] Paul D Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech house, 2013.

[27] Paul D. Groves and Ziyi Jiang. Height aiding, C/N0 weighting and consistency checking for GNSS NLOS and multipath mitigation in urban areas. *The Journal of Navigation*, 66(5):653–669, 09 2013.

[28] Paul D Groves, Lei Wang, and M Ziebart. Shadow matching: Improved GNSS accuracy in urban canyons. *GPS World*, 23(2):14–18, 2012.

[29] Li-Ta Hsu, Yanlei Gu, and Shunsuke Kamijo. NLOS correction/exclusion for GNSS measurement using RAIM and city building models. *Sensors*, 15(7):17329–17349, 2015.

[30] Irfanullah, Kamal Haider, Qasim Sattar, Sadaqat ur Rehman, and Amjad Ali. An efficient approach for sky detection. *International Journal of Computer Science Issues (IJCSI)*, 10(4):222–226, 2013.

[31] Magdy F Iskander and Zhengqing Yun. Propagation prediction models for wireless communication systems. *IEEE Transactions on microwave theory and techniques*, 50(3):662–673, 2002.

[32] Shengyue Ji, Wu Chen, Xiaoli Ding, Yongqi Chen, Chunmei Zhao, and Congwei Hu. Potential Benefits of GPS/GLONASS/GALILEO Integration in an Urban Canyon – Hong Kong. *The Journal of Navigation*, 63:681–693, 10 2010.

[33] Ziyi Jiang, Paul D Groves, Washington Y Ochieng, Shaojun Feng, Carl D Milner, and Philip G Mattos. Multi-constellation GNSS multipath mitigation using consistency checking. In *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*, pages 3889–3902. Institute of Navigation, September 2011.

[34] Helsingin kaupungin tietokeskus. Helsinki Region Infoshare. Cited: 1.2.2017, available: http://www.hri.fi/.

[35] Juraj Kostolansky. Sky detection using SLIC superpixels. Cited: 20.09.2016, available: http://vgg.fiit.stuba.sk/2015-02/sky-detection-using-slic-superpixels/.

[36] Matthew Lashley. *Modeling and performance analysis of GPS vector tracking algorithms*. PhD thesis, Auburn University, 2009.

[37] Lawrence Lau and Paul Cross. Investigations into phase multipath mitigation techniques for high precision positioning in difficult environments. *Journal of Navigation*, 60(03):457–482, 2007.

[38] Space Systems Finland Ltd. SSF website. Cited: 18.1.2017, available: http://www.ssf.fi/.

[39] Trimble Navigation Ltd. GNSS planning online. Cited: 25.11.2016, available: http://www.trimble.com/GNSSPlanningOnline/.

[40] Jun-ichi Meguro, Taishi Murata, Jun-ichi Takiguchi, Yoshiharu Amano, and Takumi Hashizume. GPS multipath mitigation for urban area using omnidirectional infrared camera. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):22–30, March 2009.

[41] Cetin Mekik and Ozer Can. Multipath effects in RTK GPS and a case study. *Journal of Aeronautics, Astronautics and Aviation, Series A*, 42(4):231–240, 2010.

[42] S. Miura, S. Hisaka, and S. Kamijo. GPS multipath detection and rectification using 3D maps. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1528–1534. IEEE, Oct 2013.

[43] Septentrio Satellite Navigation N.V. PP-SDK. Cited: 7.10.2016, available: http://www.septentrio.com/products/software/other-post-processing-software.

[44] M. Obst, S. Bauer, P. Reisdorf, and G. Wanielik. Multipath detection with 3D digital maps for robust multi-constellation GNSS/INS vehicle localization in urban areas. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 184–190. IEEE, June 2012.

[45] M. Obst, S. Bauer, and G. Wanielik. Urban multipath detection and mitigation with dynamic 3D maps for reliable land vehicle localization. In *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pages 685–691. ION, April 2012.

[46] Sébastien Peyraud, David Bétaille, Stéphane Renault, Miguel Ortiz, Florian Mougel, Dominique Meizel, and François Peyret. About non-line-of-sight satellite detection and exclusion in a 3D map-aided localization algorithm. *Sensors*, 13(1):829–847, 2013.

[47] Chris Rizos and Dorota A. Grejner-Brzezinska. Geodesy and Surveying. In Scott Gleason and Demoz Gebre-Egziabher, editors, *GNSS Applications and Methods*, chapter 14, pages 347–381. Artech House, 2009.

[48] Chris Rizos, Dru Smith, Stephen Hilla, Joe Evjen, and William Henning. GPS Projects: Some Planning Issues. In John D Bossler, John R Jensen, Robert B

McMaster, and Chris Rizos, editors, *Manual of Geospatial Science and Technology (2nd ed.)*, chapter 12, pages 191–215. CRC Press, 2010.

[49] Christian Rost and Lambert Wanninger. Carrier phase multipath mitigation based on GNSS signal quality measurements. *Journal of Applied Geodesy*, 3(2):81–87, 2009.

[50] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701. IEEE, Oct 2006.

[51] Davide Scaramuzza. *Omnidirectional vision: from calibration to robot motion estimation*. PhD thesis, ETH Zurich, 2008.

[52] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, pages 45–45. IEEE, 2006.

[53] D. Schneider, E. Schwalbe, and H.-G. Maas. Validation of geometric models for fisheye lenses. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(3):259–266, 2009.

[54] Günter Seeber. *Satellite Geodesy, 2nd completely revised and extended edition*. Walter de Gruyter, 2003.

[55] Gaurav Sharma and Raja Bala. *Digital color imaging handbook*. CRC press, 2002.

[56] E. Shytermeja, A. Garcia-Pena, and O. Julien. Proposed architecture for integrity monitoring of a GNSS/MEMS system with a fisheye camera in urban environment. In *International Conference on Localization and GNSS 2014 (ICL-GNSS 2014)*, pages 1–6. ION, June 2014.

[57] Jean-Marie Sleewaegen and Frank Boon. Mitigating short-delay multipath: a promising new technique. In *Proceedings of the 14th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2001)*, pages 204–213. ION, September 2001.

[58] Taro Suzuki and Nobuaki Kubo. N-LOS GNSS signal detection using fish-eye camera for vehicle navigation in urban environments. In *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014*, pages 1897–1906. ION, 2014.

[59] Tomoji Takasu and Akio Yasuda. Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. In *International symposium on GPS/GNSS*, pages 4–6. International Convention Centre Jeju, Korea, November 2009.

[60] PJG Teunissen and S Verhagen. GNSS ambiguity resolution: when and how to fix or not to fix? In *VI Hotine-Marussi Symposium on Theoretical and Computational Geodesy*, pages 143–148. IAG, 2008.

[61] Bryan R Townsend, Patrick C Fenton, Keith J Dierendonck, and DJ Richard Nee. Performance evaluation of the multipath estimating delay lock loop. *Navigation*, 42(3):502–514, 1995.

[62] Lei Wang, Paul D. Groves, and Marek K. Ziebart. Multi-constellation GNSS performance evaluation for urban canyons using large virtual reality city models. *The Journal of Navigation*, 65(3):459–476, 07 2012.

[63] Lei Wang, Paul D Groves, and Marek K Ziebart. GNSS shadow matching: Improving urban positioning accuracy using a 3D city model with optimized visibility scoring scheme. *Navigation*, 60(3):195–207, 2013.

[64] David Wells, Norman Beck, Alfred Kleusberg, Edward J Krakiwsky, Gerard Lachapelle, Richard B Langley, Klaus-peter Schwarz, James M Tranquilla, Petr Vanicek, and Demitris Delikaraoglou. *Guide to GPS Positioning.* Canadian GPS Associates, 1987.

[65] Oliver J Woodman. An introduction to inertial navigation. Technical Report UCAMCL-TR-696, University of Cambridge, 2007.

[66] Gerhard Wübbena, Andreas Bagge, and Martin Schmitz. Network-based techniques for RTK applications. In *GPS Symposium, GPS JIN*, pages 14–16. Japan Institute of Navigation, November 2001.